4-2019

Wavelet-Based Multi-Step Methods for Initial-Value Problem

Athar J. S Bsharat

United Arab Emirates University

College of Science

Department of Mathematical Sciences

WAVELET-BASED MULTI-STEP METHODS FOR INITIAL-VALUE PROBLEMS

Athar J. S. Bsharat

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science in Mathematics

Under the Supervision of Dr. Mohamed A. Hajji

April 2019

Declaration of Original Work

I, Athar S. Bsharat, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this thesis entitled *"Wavelet-Based Multi-Step Methods for Initial-Value Problems"*, hereby, solemnly declare that this thesis is my own original research work that has been done and prepared by me under the supervision of Dr. Mohamed A. Hajji, in the College of Science at UAEU. This work has not previously been presented or published, or formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my thesis have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this thesis.

Student's Signature Ath Date 16/5/2019.

Approval of the Master Thesis

This Master Thesis is approved by the following Examining Committee Members:

1) Advisor (Committee Chair): Dr. Mohamed A. Hajji

Title: Associate Professor

Department of Mathematical Sciences

College of Science

Signature _____ Date April 18, 2019

2) Member: Dr. Youssef EL-Khatib

Title: Associate Professor

Department of Mathematical Sciences

College of Science

Signature _____ Date April 18, 2019

3) Member (External Examiner): Dr. Hishyar Khalil Abdullah

Title: Associate Professor

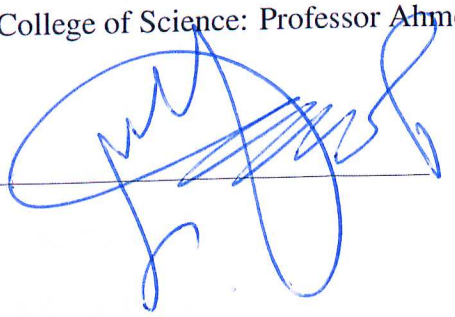
Department of Mathematics

Institution: University of Sharjah

Signature _____ Date 18/4/2019

This Master Thesis is accepted by:

Dean of the College of Science: Professor Ahmed Murad

Signature  Date 16/5/2019

Acting Dean of the College of Graduate Studies: Professor Ali Al-Marzouqi

Signature Ali Hassan Date 19/5/2019

Copy 1 of 8

Abstract

Wavelets has been a popular tool, since the late 1980s in many areas of engineering and mathematics. A major contribution of wavelets is their adaptation in the JPEG2000 picture format standard in 2000 and in the compression and storage of finger print scans. Since then wide applications of wavelets in different areas have emerged. Popular wavelets are the compactly-supported wavelets constructed by I. Daubechies. In this work, we use Daubechies' wavelets to develop multistep algorithms for the solution of initial value problems (IVPs) in the context of Galerkin method. Though, such wavelet basis functions have good approximation property, they do not have explicit formulae, making finding inner products a challange. This work tackles this point and uses the order of approximation of the wavelets to derive implicit multistep methods with comparable stability property to other methods. The derived methods are tested on linear and non-linear test equations.

Keywords: Wavelet Basis, Daubechies' Wavelets, Multiresolution Analysis, Multistep methods, Initial Value Problems, Galerkin Method.

Title and Abstract (in Arabic)

طرق متعددة الخطوات لمعادلات القيمة البدائية اعتمادا على المويجات

الملخص

تعد المويجات أداة شائعة منذ أواخر الثمانينات في العديد من مجالات الهندسة والرياضيات. تتمثل إحدى الإسهامات الرئيسية للمويجات في تنسيق صور (چعيج٢٠٠٠) في عام ٢٠٠٠ وفي ضغط وتخزين عمليات مسح بصمات الأصابع. ومنذ ذلك الحين ظهرت تطبيقات عدة للمويجات في مجالات مختلفة. المويجات الأكثر شيوعاً هي التي تم بناؤها من قِبل ديوبشيس والتي تتميز بدعم مدمج. في هذه الإطروحة ، نستخدم موجات ديوبشيس لتطوير خوارزميات متعددة الخطوات لحل معادلات القيمة الأولية في سياق طريقة جليركن. على الرغم من أن الدوال الأساسية للمويجات لها خاصية تقريب جيدة، إلا أنها لا تحتوي على صيغ صريحة، ويشكل هذا بالتأكيد تحدياً في ايجاد المنتج الداخلي. يعالج هذا الأمر بهذه الأطروحة ويستخدم رتب تقريب المويجات لاشتقاق طرق متعددة الخطوات الضمنية مع خاصية ثبات قابلة للمقارنة مع طرق أخرى. يتم اختبار الطرق المشتقة على معادلات الاختبار الخطية وغير الخطية.

مفاهيم البحث الرئيسية : أساس المويجات، مويجات ديوبشيس ، تحليل متعدد الدرجات ، طرق متعددة الخطوات، معادلات القيمة الأولية ، طريقة جليركن .

Acknowledgements

During my journey for studying Mathematics in the master program, proposed by the United Arab Emirates University, I have been promoted, affirmed, and enlivened by many people whom without them standing by me, it would not have been possible to fill out the course of study.

My deepest gratitude go to my supervisor, Associate Professor Mohamed A. Hajji, for his priceless support, patience and understanding, with which I was able to overpass all the troubles in my inquiry. I would also like to thank Dr. Mohamed Salim, the head of the Department of Mathematical Sciences, and all of my Professors at the department for their support. My sincere appreciations also go to Dr. Hishyar Khalil Abdullah from the University of Sharjah, the external examiner, for being part of the examination committee.

I would like to thank my beloved family, beginning with my parents, whom without their love and encouragement, I would have never been where I am today. My thanks, with love, also go to my husband and kids for their patience and support throughout this long journey.

In addition, special thanks are extended to my friend Azza for her assistance and supporting.

Dedication

To my beloved family and teachers

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Wavelets are special mathematical functions that satisfy certain mathematical requirements. Wavelets were developed in the fields of mathematics, physics, engineering, and seismic geology. This led to a various applications as in numerical analysis, image and video compression, fingerprints, speech recognition and earthquakes predictions [13].

The historical origins of the wavelets date back to the beginning of the 20th century, when Haar [2] constructed the first and simplest wavelet function in 1901. In 1986, Mallat [3] and Meyer [4] introduced the concept of a multiresolution analysis (MRA) which give a natural framework of wavelet approximation and the construction of orthonormal wavelet basis .

In 1987, Daubachies [5, 6] constructed the first ever known family of orthogonal wavelets with compact support. Since then numerous applications of wavelets were found. One of the successes of wavelets has been in signal processing [7, 8]. Wavelets have also been used in the numerical solution of differential equations. For example, Chen and Hsiao [9] had derived an operational matrix of integration based on Haar wavelet, Lepik [10] had used Haar wavelets to solve higher order as well as nonlinear ODEs. Xu and Shann [12] have used wavelet-based Galerkin methods to solve two-point boundary problems.

Differential equations are one of the most mathematical tools that have a wide range of applications in physics, biology and engineering, in particular initial value problems which generally describe the evolution in time of the state of a system. It is well-known that many initial value problems do not admit exact solutions. Over the past decades, many numerical methods to approximate the solution of initial value problems have been constructed. Popular numerical methods are Euler method, the finite difference methods and Rung-Kutta methods [1]. Other methods known as spectral methods have been developed as well and are widely used [14–19]. As mentioned

above wavelets have also been used for the numerical solution of differential equations However, many of these works use either the Haar wavelets [10, 20, 21], or polynomial like wavelets [22–25].

In this thesis we use Daubechies' wavelets [5] in a Galerkin setting to derive multistep methods to solve first order initial value problems $y' = f(t,y)$. Daubechies wavelets are known for their good approximation of order p in the sense that polynomials of degree $p-1$ are exactly generated by linear combination of the integral translates of the scaling functions. This nice property will be used throughout to guarantee that the order of the derived method is $O(h^p)$. The resolution, m, of the approximation spaces will determine the step size of the method, $h = 2^{-m}$.

The thesis is organized as follows. In Chapter 2, an overview of wavelet theory is provided a long with needed results pertaining to the derivation of the multistep methods. We first introduce the wavelet transform and comparison with the Fourier transform. Next , we give the definition of the multiresolution analysis (MRA) and the scaling functions, review the main ingredients of orthonormal wavelet bases and point out the properties used later in this work. Using the multiresolution analysis concept, we highlight the construction of Daubechies' compactly-supported wavelets. In the end of Chapter 2, we describe an algorithm which can be used to compute the exact values of Daubechies' compactly-supported scaling and wavelets functions as well as quadratures approximations for the scaling coefficients of a function.

In Chapter 3, we describe the derivation of implicit wavelet-based multistep schemes for the numerical solution of first-order initial value problems $y' = f(t,y)$, $y(t_0) = y_0$. First, we describe the general approach to derive the wavelet Galerkin method (WGM). Then, based on the approximation property of Daubechies wavelets, a modification of this WGM, is described leading the modified wavelet Galerkin method of order p (MWGMp). We will modify the WGM in three different ways leading to three different stable multistep methods.

In Chapter 4, we discuss the order of convergence and the stability of the three different modified wavelet Galerkin multistep methods. In Chapter 5, we present some numerical examples to verify the accuracy of the derived methods. Finally, in Chapter 6, a conclusion is drawn and future work is suggested.

Chapter 2: An Overview of Wavelets

In this Chapter, we present an overview of wavelet theory. In particular, we present results pertaining to compactly-supported wavelets constructed by I. Daubechies [5]. We present results and properties of such wavelets that are needed in later chapters dealing with the construction of multistep methods for solving initial value problems. There are many references in the literature about wavelets and the reader can refer to for example [26–33] and references therein.

2.1 From Fourier Transform to Wavelet Transform

Wavelets are a special mathematical functions that are used to represent and to analyse other functions (or data) at different scales. Essentially, wavelets take a function and break it down into pieces, each piece “lives" at a certain scale. The analysis of of the original function is done on each piece and then reconstruction of the original function is performed.

In fact, Approximation using the superposition of functions is not new idea. Many techniques are used to express a function as a linear combination of other basis functions such as Taylor series, the Fourier series and the Fourier transform. The latter is a popular transform used in signal analysis, where signals are transformed into frequency domain.

For any function $f \in L^1(R)$, the continuous Fourier transform and its inverse are define by

$$(\mathscr{F} f)(w) = \hat{f}(w) = \int_{-\infty}^{\infty} f(t) e^{-itw} dt, \tag{2.1}$$

$$(\mathscr{F}^{-1} f)(t) = f(t) = \int_{-\infty}^{\infty} \hat{f}(w) e^{itw} dt, \tag{2.2}$$

where w is the frequency variable. Note that the Fourier transform of $f(t)$ is a complex function. It gives the information of to what extent a certain frequency w is present

in the signal $f(t)$. This is examined by looking at the modulus $|\hat{f}(w)|$. If $|\hat{f}(w)| = 0$ (or very small), this means that frequency w is not present in the signal (or has very small contribution to the signal). On the other hand, if $|\hat{f}(w)|$ is large, this means that frequency w is present in the signal and has considerable contribution. The Fourier transform is a representation of the original function as a superposition of sines and cosines ($e^{iwt} = \cos(wt) + i\sin(wt)$). For more information about the Fourier transform, the reader can refer to [34–37].

The Fourier transform as defined in (2.1) has two main problems. First, it is unable to localise in time when a ceratin frequency happened, i.e., even if $|\hat{f}(w^*)|$ is large, we would not know at what time the frequency w^* occurred. This is due to the non-compact support of the trigonometric functions sine and cosine. Another inconvenience is that we require $f(t)$ for all t to compute $\hat{f}(w)$ at a particular w. To partially overcome these two drawbacks, the windowed-Fourier transform (WFT) was defined as a solution to localize frequency in time [38, 39].

The WFT of a function $f(t)$ is defined by

$$(\mathscr{W}\mathscr{F}f)(w,t) = \int_{-\infty}^{\infty} f(s)g(s-t)e^{-iwt}ds, \tag{2.3}$$

where $g(x)$ is called the window function, a function with compact support. For example,

$$g(t) = \begin{cases} 1, & 0 \leq t \leq \Delta, \\ 0, & \text{otherwise}, \end{cases} \tag{2.4}$$

where Δ is the width of the window. The WFT computes the frequency content of the signal function $f(t)$ in a neighborhood of a point in time t. Precisely, $(\mathscr{W}\mathscr{F}f)(w,t)$ gives the frequency content of the signal in the interval in time $[t, t+\Delta]$. It chops up the function $f(t)$ into pieces, each piece defined on $[t, t+\Delta]$, and then analyzes each

piece for its frequency content by taking the Fourier transform. We can see that the WFT solves the time-frequency localization problem. However, the WFT still has a disadvantage which is in its window function $g(s,t)$. It can be seen that $g(s,t)$ always has the same width Δ.

This problem of constant width of the window function $g(t)$ is solved by the introduction of another transform called the wavelet transform. The continuous wavelet transform of a function $f(x) \in L^2(R)$ is defined by

$$(Wf)(a,b) = \int_{-\infty}^{\infty} f(x)\psi_{a,b}(t)dt, \tag{2.5}$$

where the doubly-indexed functions $\psi_{a,b}(t)$ are defined by

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \qquad a,b \in R,\ a \neq 0, \tag{2.6}$$

and $\psi(t)$ is the mother wavelet. It can be seen that all the functions $\psi_{a,b})(t)$, called wavelet functions, are derived from the mother wavelet $\psi(t)$ by means of dilation and translation. The variable b in $(Wf)(a,b)$ no longer represents frequency. It represents "scale". The variable a is a time localization variable. For the wavelet transform (2.5) to be invertible, the mother wavelet function $\psi(t)$ has to satisfy the admissibility condition [6].

$$C_\psi := 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\xi)|^2}{\xi} d\xi < \infty, \tag{2.7}$$

where $\hat{\psi}(\xi)$ is the Fourier transform of $\psi(t)$.

If $\psi(t)$ satisfies equation (2.7), then the wavelet transform (2.5) is invertible on its range,and $f(t)$ can be recovered by

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} (Wf)(a,b)\psi_{a,b}(t)\frac{dadb}{a^2} \tag{2.8}$$

The admissibility condition (2.7) implies the necessary condition that $\widehat{\psi}(0) = 0$ which means that $\int_{-\infty}^{\infty} \psi(t)dt = 0$. Thus, $\psi(t)$ is oscillating and decays to zero at $\pm\infty$. This oscillation property, together with the decay property, has given $\psi(t)$ the name wavelet.

In practice, a discrete version of (2.5) is used, the discrete wavelet transform (DWT). In the DWT, the dilation and translation parameters a and b are restricted to discrete values $a = a_0^{-j},\ b = kb_0a_0^{-j},\ k, j \in Z$, with $a_0 > 1, b_0 > 0$. The resulting discrete transform of (2.5) is then

$$(Wf)(j,k) := a_0^{j/2} \int_{-\infty}^{\infty} f(t)\psi(a_0^j t - kb_0)dt. \tag{2.9}$$

For the particular values $a_0 = 2$ and $b_0 = 1$, the widely used DWT is

$$(Wf)(j,k) := 2^{j/2} \int_{-\infty}^{\infty} f(t)\psi(2^j t - k)dt. \tag{2.10}$$

In concluding this section, it is worth noting that the wavelet transform is similar to WFT in that they both provide a time-frequency analysis of the function $f(t)$. The difference between them is that in WFT the support of the window functions $g(s-t)$ remain fixed for all frequencies. However, for the wavelet transform, the window functions are $\psi_{a,b} = \frac{1}{\sqrt{a}}\left(\frac{x-b}{a}\right)$ which have support depending on the scale (frequency) a. This type of dependence makes the wavelet transform better in detecting short-lived high variations (high frequency) in the function $f(t)$. Another approach to understand wavelets is through the concept of multiresolution analysis [3] discussed in the next section.

2.2 The Multiresolution Analysis and Wavelets

In this section, we introduce wavelets (the discrete wavelet transform) through the concept of multiresolution analysis (MRA). We will see how orthonormal wavelet basis of $L^2(R)$ are constructed using the MRA. In the first subsection, we define the

MRA and the construction of the *scaling function* $\phi(x)$. In the second subsection, we define the wavelet spaces and the mother wavelet function $\psi(x)$.

2.2.1 The Multiresolution Analysis

The multiresolution analysis is defined as follows.

Definition 2.2.1. A one-dimensional orthonormal multiresolution analysis $(1-D\,MRA)$ of $L^2(R)$ is defined as increasing sequence of closed subspaces V_j of $L^2(R)$, $j \in Z$, such that

$$0 \subset \ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots \subset L^2(R) \tag{2.11}$$

with the following properties.

1. $\bigcap_{j\in Z} V_j = 0$ and $\bigcup_{j\in Z} V_j$ is dense in $L^2(R)$.
2. $f(x) \in V_j \; iff f(2x) \in V_{j+1}$, for all $j \in Z$.
3. $f(x) \in V_j \; iff f(x-2^{-j}k) \in V_j$, for all $k \in Z$.
4. There exist a function $\phi(x) \in V_0$ with non-vanishing integral such that the set $\{\phi_{0,k}(x) = \phi(x-k), k \in Z\}$ is an orthonormal basis of V_0.

The function $\phi(x)$ is called the mother scaling function. From the definition, it is easy to see that for any $j \in Z$, the family

$$\phi_{j,k}(x) = 2^{\frac{j}{2}}\phi(2^j x - k), \qquad k \in Z, \tag{2.12}$$

is an orthonormal basis of V_j. Since $\phi(x) \in V_0 \subset V_1$, there exist a sequence $\{h_k, k \in Z\}$ such that

$$\phi(x) = \sum_{k\in Z} h_k \phi_{1,k}(x) = \sqrt{2}\sum_{k\in Z} h_k \phi(2x-k). \tag{2.13}$$

The equation (2.13) is known by different names such as the dilation equation, the two-scale difference equation, or the refinement equation. The coefficients h_k, in engineering applications, are called the low pass filter coefficients. They satisfy the following properties whose proofs can be found in many wavelet textbooks such as [6].

(P1) $$\sum_{k\in Z} h_k = \sqrt{2}. \tag{2.14}$$

(P2) $$\sum_{k\in Z} h_k^2 = 1. \tag{2.15}$$

(P3) $$\sum_{k\in Z} (-1)^k h_k = 0. \tag{2.16}$$

(P4) $$\sum_k h_{2k} = \frac{\sqrt{2}}{2} = \sum_k h_{2k+1}. \tag{2.17}$$

The scaling function $\phi(x)$ is uniquely defined by its refinement equation (2.13) and the normalization

$$\int_{-\infty}^{\infty} \phi(x)dx = c, \text{ (a constant)}.$$

The MRA spaces V_j are called the approximation spaces and are used to approximate functions of $L^2(R)$. Given a function $f(x) \in L^2(R)$, it is approximated by its orthogonal projection, f_j, onto V_j,

$$f(x) \approx P_j(f)(x) = f_j(x) = \sum_{k\in Z} s_{j,k}\phi_{j,k}(x), \tag{2.18}$$

with the coefficients $s_{j,k}$, called the scaling coefficients, are given by

$$s_{j,k} = \int_{-\infty}^{\infty} f(x)\phi_{j,k}(x)dx. \tag{2.19}$$

To be able to express the simplest constant functions $f(x) = c$ as a linear combination of the collection $\phi(x-l), l \in Z$, it can be easily verified that the scaling function $\phi(x)$ satisfies

$$\sum_l \phi(x-l) = constant. \tag{2.20}$$

If the normalization is $\int_{-\infty}^{\infty}\phi(x)dx = 1$, we have $\sum_l \phi(x-l) = 1$. In many cases, there is no available explicit expression for $\phi(x)$, it is defined by its refinement equation (2.13) and the normalization (2.20). However, as we will see in Section 2.4, there is an algorithm to compute the values of the scaling functions $\phi(x)$ at dyadics points $\{x = \frac{k}{2^j},\ j,k \in Z\}$.

Although $\phi(x)$ does not have an explicit expression in x domain, its Fourier transform $\hat{\phi}(\psi)$ can be expressed formally in a compact form as follows. Taking the Fourier transform of (2.13), we obtain

$$\hat{\phi}(\xi) = m_0(\xi/2)\hat{\phi}(\xi/2), \tag{2.21}$$

where

$$m_0(\xi) = \frac{1}{\sqrt{2}}\sum_k h_k e^{-ik\xi} \tag{2.22}$$

is a 2π-periodic trigonometric function, the discrete Fourier transform of the filter coefficients h_k. Repeated application of (2.21), gives the formal expression for $\hat{\phi}(\xi)$:

$$\hat{\phi}(\xi) = \prod_{j=1}^{\infty} m_0(\xi/2^j). \tag{2.23}$$

The infinite product converges if and only if $m_0(0) = 1$ which means that $\sum_k h_k = \sqrt{2}$, a property we mentioned before. Moreover, the orthonormality of $\{\phi(x-k),\ k \in Z\}$ is equivalent to

$$|m_0(\xi)|^2 + |m_0(\xi+\pi)|^2 = 1, \qquad \forall\ \xi. \tag{2.24}$$

2.2.2 The Detail Spaces and The Wavelet Functions

In this section we will introduce the wavelets functions and the wavelet basis for $L^2(R)$. Given an multiresolution analysis, as defined in the previous subsection, for

each $j \in Z$, let W_j be the orthogonal complement of V_j in $V_j + 1$. Then we have

$$V_{j+1} = V_j \oplus W_j \qquad \text{and} \qquad W_j \perp W_{j'} \quad j \neq j' \tag{2.25}$$

where the symbol $\oplus$ denote the direct sum. Equation (2.25) means that each elements in V_{j+1} can be written uniquely as a sum of an element of V_j and an element of W_j. The spaces W_j contain the information that we need to move from a given approximation space V_j with a resolution j to another approximation space V_{j+1}. These spaces are referred to the “detail spaces" or wavelet spaces.

The idea behind the multiresolution analysis is that whenever we have a scaling function $\phi(x)$ associated with a multiresolution analysis, there exist a wavelet function $\psi(x) \in W_0$, called the mother wavelet function, such that the set $\{\psi_{0,k} = \psi(x-k), k \in Z\}$ is an orthonormal basis of the detail space W_0, and the family

$$\psi_{j,k} = 2^{j/2}\psi(2^j x - k), \qquad k \in Z, \tag{2.26}$$

is an orthonormal basis for W_j. The set $\{\psi_{j,k},\ j,k \in Z\}$ is the orthonormal *wavelet basis* of $L^2(R)$ associated with the multiresolution analysis. The mother wavelet $\psi(x)$ is given by [5,6]

$$\psi(x) = \sqrt{2}\sum_k g_k \phi(2x-k), \tag{2.27}$$

with $g_k = (-1)^k h_{1-k}$.

2.3 Daubechies' Compactly-Supported Wavelets

The wavelet bases discussed in previous section consist of functions that are supported on the entire real line. However, in many applications, it is desirable to work with wavelets with compact support. In this section, we shall describe the construction of I. Daubechies [5,6]. These wavelets are not only compactly supported but they have good approximation property.

As we saw in the previous section, all depend on the scaling function $\phi(x)$, which is a solution of the refinement equation

$$\phi(x) = \sqrt{2}\sum_{k} h_k \phi(2x-k). \tag{2.28}$$

In her construction of compactly supported ϕ, Daubechies seeked a solution of the finite refinement equation:

$$\phi(x) = \sqrt{2}\sum_{k=0}^{N-1} h_k \phi(2x-k). \tag{2.29}$$

such that $\{\phi(x-k),\ k \in Z\}$ is an orthonormal basis for V_0. This orthonormality property as we saw in the previous section is equivalent to

$$|m_0(\xi)|^2 + |m_0(\xi+\pi)|^2 = 1, \tag{2.30}$$

where $m_0(\xi)$ be comes the 2π-periodic trigonometric polynomial

$$m_0(\xi) = \frac{1}{\sqrt{2}}\sum_{k=0}^{N-1} h_k e^{-ik\xi}. \tag{2.31}$$

Daubechies solved for $m_0(\xi)$ satisfying the orthonormality condition (2.30) and having the form

$$m_0(\xi) = \left(\frac{e^{i\xi}+1}{2}\right)^p \mathscr{L}(\xi), \qquad p \geq 1, \tag{2.32}$$

where $\mathscr{L}(\xi)$ is a trigonometric polynomial. For more details on Daubechies' construction, the reader is encouraged to see [6].

It has been proven in [6] that $\phi(x)$ in (2.29) is compactly-supported in $[0, N-1]$ and the corresponding mother wavelet function $\psi(x)$, after translation, is given by

$$\psi(x) = \sqrt{2}\sum_{k=0}^{N-1} g_k \phi(2x-k), \tag{2.33}$$

and is also compactly-supported in $[0, N-1]$, where g_k are given by

$$g_k = (-1)^k h_{N-k-1}, \qquad k = 0, 1, \ldots, N-1. \tag{2.34}$$

The construction of $m_0(\xi)$ satisfying (2.32) implies that $\phi(x)$ has p^{th} order approximation property, in the sense that any polynomial, $P_r(x)$, of degree less r than or equal to $p-1$ can be generated exactly by integral translates of $\phi(x)$, i.e.,

$$P_r(x) = \sum_k c_k \phi(x-k), \qquad 0 \le r \le p-1. \tag{2.35}$$

This property translates to the wavelet function $\psi(x)$ satisfying p vanishing moments:

$$\int_{-\infty}^{\infty} x^r \psi(x) dx = 0, \qquad 0 \le r \le p-1. \tag{2.36}$$

The above mentioned approximation property of the scaling function $\phi(x)$ will be very useful in the next chapter in constructing the multistep algorithm for solving initial value problems. As we saw, Daubechies' scaling function does not have a closed form expression. In numerical calculations, one needs the values of $\phi_{j,k}(x)$. In the next section, we describe an algorithm, described in details in [40], by which $\phi(x)$ can be calculated at dyadic points $x = \frac{k}{2^j},\ k, j \in Z$.

2.4 The Scaling and Wavelet Functions Values

In this section we will describe an algorithm [40] to compute the *exact* values of a compactly supported scaling functions $\phi_{j,k}(x)$ and $\psi_{j,k}(x)$. Since all of the scaling and wavelet functions $\phi_{j,k}(x)$ and $\psi_{j,k}(x)$ are given in terms of the mother scaling function $\phi(x)$ (see Eqs. (2.29) and (2.33)), it suffices to consider $\phi(x)$ at dyadic points $x = \frac{k}{2^j},\ k, j \in Z$.

Let $\phi(x)$ be a compactly supported scaling function satisfying the refinement

equation

$$\phi(x) = \sqrt{2}\sum_{k=0}^{N-1} h_k \phi(2x-k) \tag{2.37}$$

with

$$\sum_{k=0}^{N-1} h_k = \sqrt{2}. \tag{2.38}$$

ϕ at the integers $x = 1,2,\ldots,N-2$

Since $\phi(x)$ is compactly supported in $[0,N-1]$, $\phi(x) = 0$ for $x \leq 0$ and for $x \geq N-1$. Let $\vec{\phi}^{(0)} = [\phi(1)\ \phi(2)\ \cdots\ \phi(N-2)]^t$ be the column vector containing ϕ at the integers $x = 1,2,\ldots,N-2$. Then evaluating (2.37) at the integers, we obtain the system

$$\vec{\phi}^{(0)} = A\vec{\phi}^{(0)} \tag{2.39}$$

where A is an $(N-2)\times(N-2)$ matrix with entries a_{ij} given by

$$\begin{cases} a_{ij} = \sqrt{2}h_{2i-j}, & \text{if } 0 \leq (2i-j) \leq N-1, \\ 0, & \textit{otherwise}, \end{cases} \tag{2.40}$$

Equation (2.39) suggests that $\vec{\phi}^{(0)}$ is an eigenvector of A corresponding to the eigenvalue $\lambda = 1$. The existence of $\lambda = 1$ as an eigenvalue of A has been shown in [40]. Since $\phi(x)$ satisfies the normalization

$$\sum_{l} \phi(x-l) = 1, \qquad \forall\, x, \tag{2.41}$$

we have that $\sum_{l\in Z} \phi(l) = \sum_{l=1}^{N-2} \phi(l) = 1$. Therefore, the vector $\vec{\phi}^{(0)}$ is equal to the normalised eigenvector of A corresponding to the eigenvalue $\lambda = 1$. This gives the values of $\phi(x)$ at the integers $x = 1,2,\ldots,N-2$.

ϕ at the half- integers $x = n/2\ n \in Z$

Once the values of $\phi(x)$ at the integers are known, we apply the refinement equation

(2.37) with $x = n/2,\ n \in Z$, to find the values of $\phi(x)$ at the half integers,

$$\phi(n/2) = \sqrt{2}\sum_{k=0}^{N-1} h_k \phi(n-k), \qquad n = 1,2,\ldots,2(L-2)+1. \tag{2.42}$$

Equation (2.42) can be viewed as a convolution of the vector $h = \sqrt{2}[h_0, h_1, \ldots, h_{N-1}]$ with the vector $\phi^{(0)} = [\phi(1), \phi(2), \ldots, \phi(N-2)]$. The convolution of two vectors $x = [x_1, x_2, \ldots, x_n]$ and $y = [y_1, y_2, \ldots, y_m]$, denoted by $z = x \star y$, is defined by the vector $z = [z_1, z_2, \ldots, z_{n+m-1}]$ of length $n+m-1$ given by

$$z_k = \sum_{l=\max(1,k-m+1)}^{\min(n,k)} x_l y_{k-l+1} = \sum_{l=\max(1,k-n+1)}^{\min(m,k)} x_{k-l+1} y_l. \tag{2.43}$$

ϕ at the odd dyadic values $x = n/2^j\ n \in Z, odd$.

Let $\phi^{(j)}, j \gneq 1$, be the vector of length $2^{j-1}(N-1)$ containing the values of the scaling function at the odd j-level dyadics, i.e.,

$$\phi^{(j)} = [\phi(1/2^j), \phi(3/2^j), \ldots, \phi((2^j(N-1)-1)/2^j)]^t. \tag{2.44}$$

From the refinement equation (2.37), we get

$$\phi^{(j)} = conv(\tilde{h}^j, \phi^{(j-1)}), \qquad j \gneq 2, \tag{2.45}$$

where $\tilde{h}^j$ is given by

$$\begin{cases} \tilde{h}^j_k = \sqrt{2}\, h_m, & \text{if } k = m2^{j-2},\ 0 \lneq m \lneq N-1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.46}$$

The length of the vector $\tilde{h}^j$ is equal to $2^{j-2}(L-1)+1$.

The algorithm is summarized as follows.

1. Compute the vector $\phi^{(0)}$ containing the values of ϕ at the integers as the normalized eigenvector of A corresponding to the eigenvalue $\lambda = 1$.

2. Convolve $\phi^{(0)}$ with the vector h . This gives the values of ϕ at the half integers.

3. For $j \gtrsim 2$, compute the values of ϕ at the odd j-level dyadics by convolving of the vector $\tilde{h}^j$ with the vector ϕ_{j-1}, where ϕ_{j-1} is the vector containing only the values of ϕ at the odd $(j-1)$-dyadics.

The above algorithm has been used to plot Daubechies' scaling and wavelet functions for different various of p, the number of vanishing moments of the wavelet, where $N = 2p$. Figures 2.1-2.4 display ϕ and ψ for $p = 3, 5, 7$ and 9.

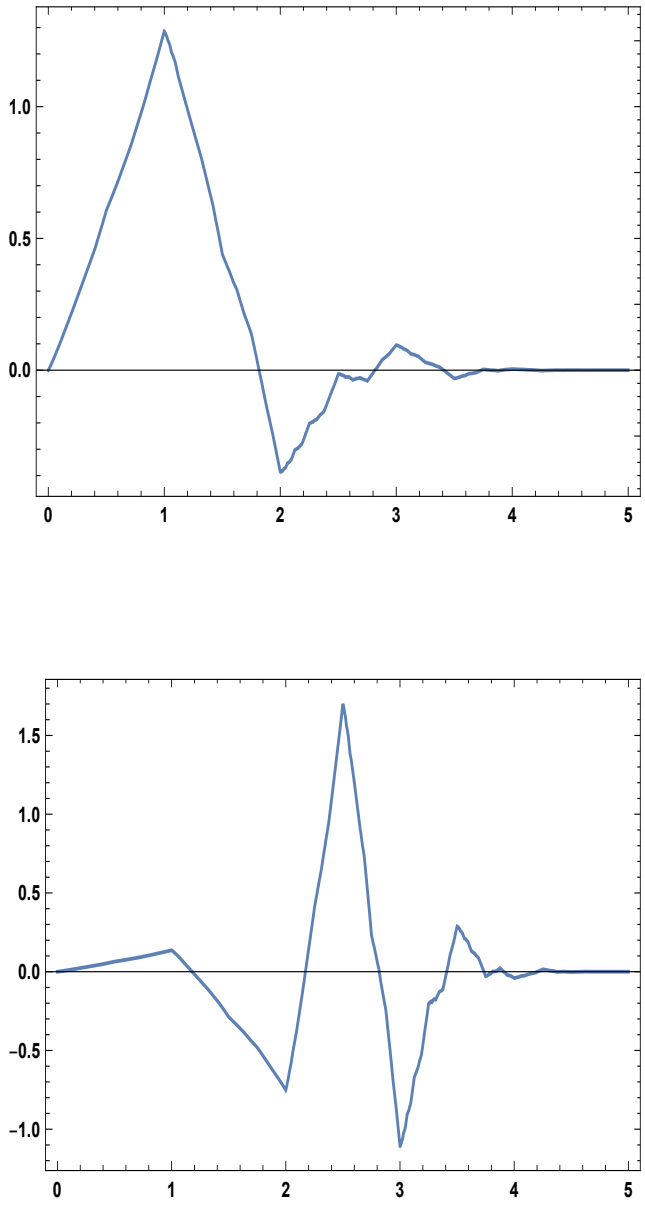
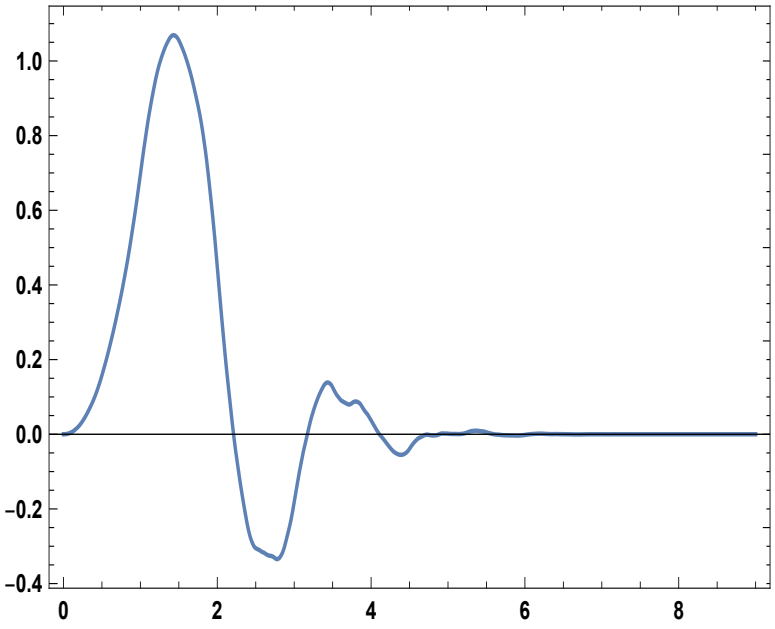


Figure 2.1: Daubechies ϕ (top) and ψ (bottom) for $p = 3$



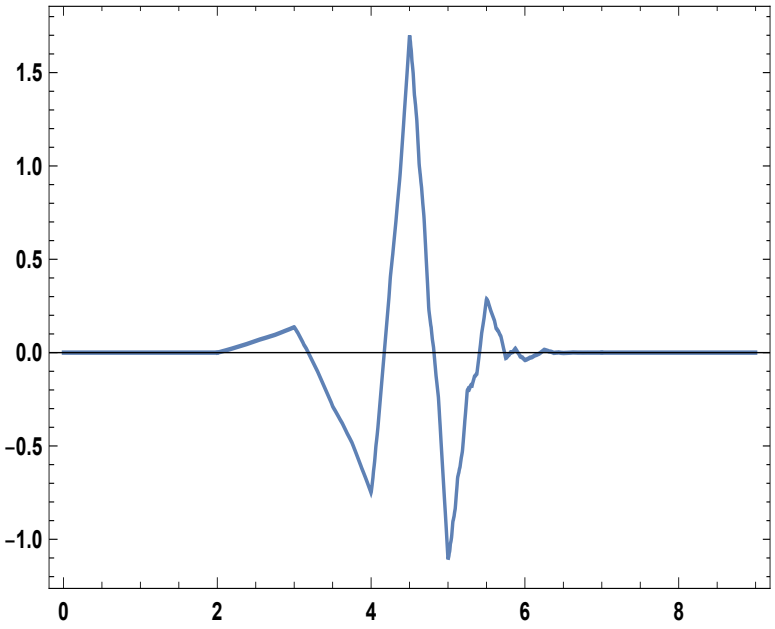


Figure 2.2: Daubechies ϕ (top) and ψ (bottom) for $p = 5$

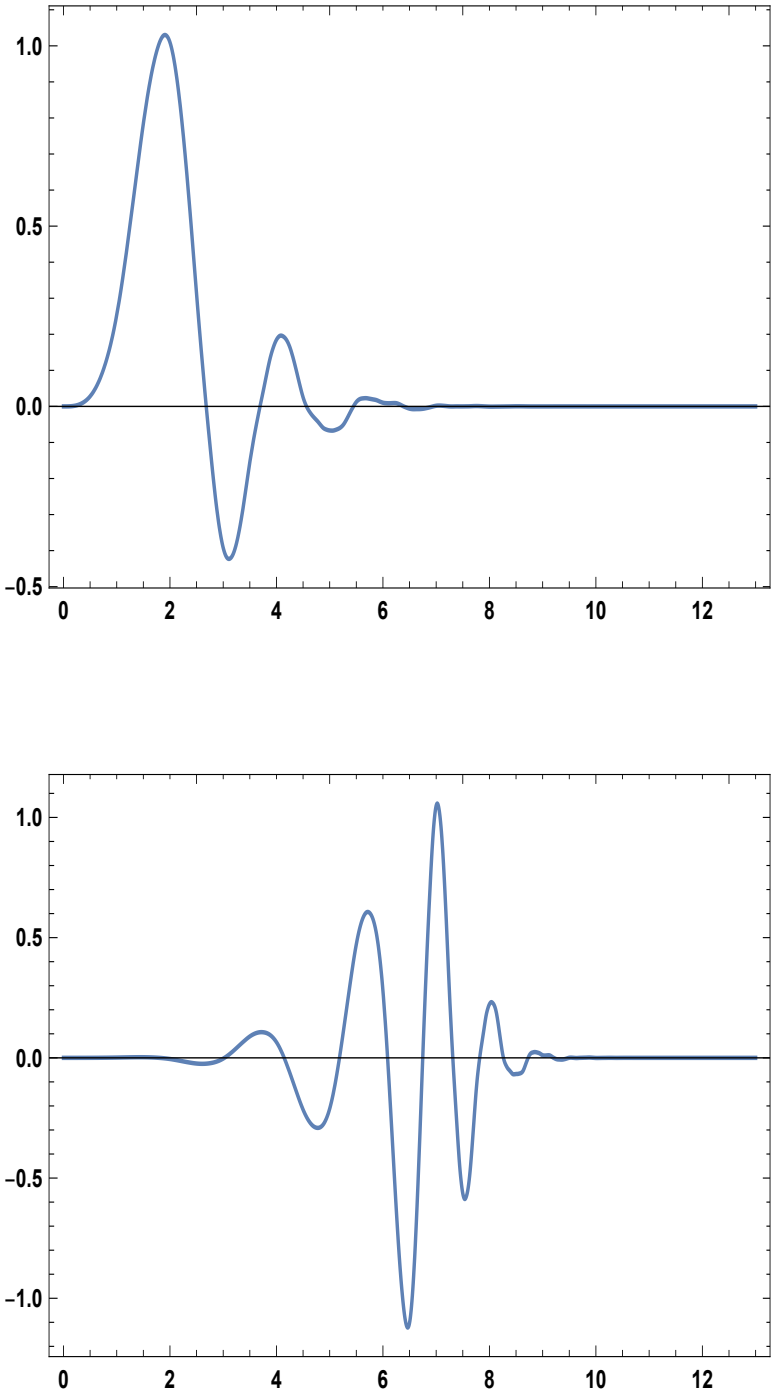


Figure 2.3: Daubechies ϕ (top) and ψ (bottom) for $p = 7$

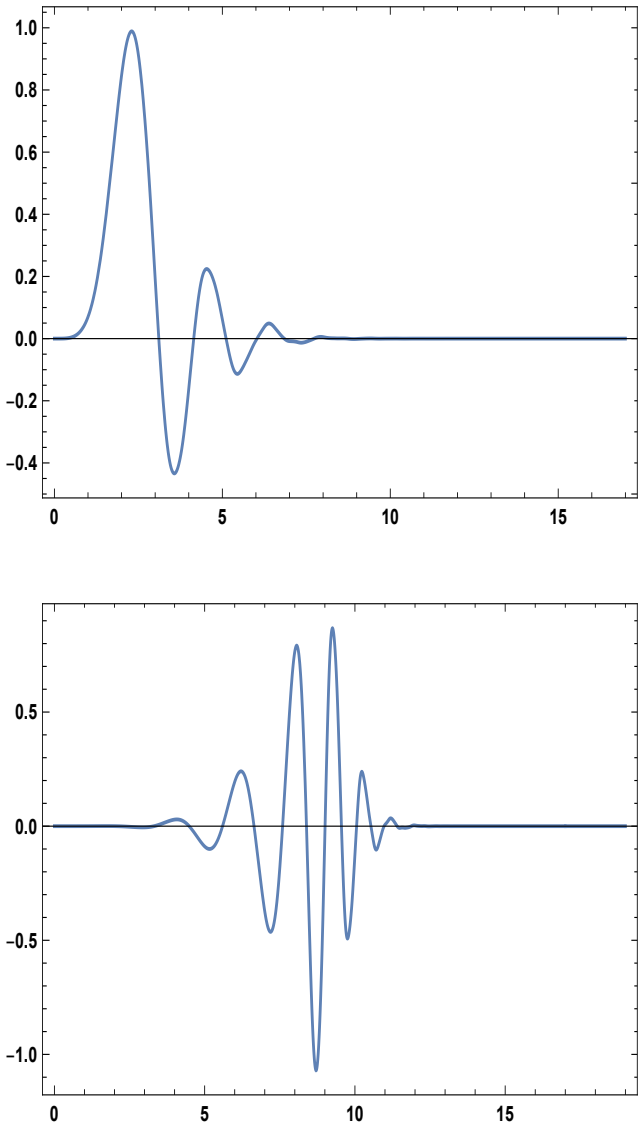


Figure 2.4: Daubechies ϕ (top) and ψ (bottom) for $p = 9$

2.5 Functions Approximation

In this section, we show how functions of $L^2(R)$ are approximated in the approximation spaces V_j. In particular, we show how one can approximate the scaling coefficients $s_{j,k}$.

Let $f(x) \in L^2(R)$ and let V_j be an approximation space and W_j be the orthogonal complement of V_j in V_{j+1}. The function f is approximated by its orthogonal projection

onto V_j:

$$f(x) \approx (P_j f) = \sum_{k \in Z} s_{j,k} \phi_{j,k}(x), \tag{2.47}$$

with the scaling coefficients $s_{j,k}$ are given by

$$s_{j,k} = \langle f, \phi_{n,k}(x) \rangle = \int_{-\infty}^{\infty} f(x) \phi_{j,k}(x) dx. \tag{2.48}$$

As $\phi_{j,k}(x)$ do not have an exact formula, one needs to approximate the integrals (2.48). In subsections 2.5.2 and 2.5.3, we present two ways to approximate $s_{j,k}$, one using a one point-quadrature (Subsection 2.5.2) and another using a multi-point quadrature (Subsection 2.5.3). In the next subsection, we present a formula for the calculation of the scaling function moments, needed later in subsections 2.5.2 and 2.5.3.

2.5.1 Moments of the Scaling Functions

In this section, we discuss the exact computation of the so-called the kth moments, μ_j^k, of the scaling functions $\phi(x-j)$. These moments will be very important in the approximation of the scaling coefficients as well as in the derivation of the schemes in Chapter 3.

The kth moment of the scaling function $\phi(x-j)$ is defined by

$$\mu_j^k = \langle x^k, \phi(x-j) \rangle = \int_{-\infty}^{\infty} x^k \phi(x-j) dx. \tag{2.49}$$

Making the change of variable $x \to x-j$, we get

$$\mu_j^k = \int_{-\infty}^{\infty} (x+j)^k \phi(x) dx = \sum_{i=0}^{k} \binom{k}{i} j^{k-i} \int_{-\infty}^{\infty} x^i \phi(x) dx = \sum_{i=0}^{k} \binom{k}{i} j^{k-i} \mu_0^i. \tag{2.50}$$

Equation (2.50) gives a formula for calculating μ_j^k in terms of μ_0^i, $i = 0, 1, \ldots, k$. It remains to find a formula to compute μ_0^i.

First, note that $\mu_0^0 = \int_{-\infty}^{\infty} \phi(x)dx = 1$. For $n \geq 1$, we have

$$\mu_0^n = \int_{-\infty}^{\infty} x^n \phi(x)\, dx. \tag{2.51}$$

Using the definition of $\phi(x)$, $\phi(x) = \sqrt{2}\sum_k h_k \phi(2x-k)$, we have

$$\begin{aligned}
\mu_0^n &= \sqrt{2}\sum_k h_k \left[\int_{-\infty}^{\infty} x^n \phi(2x-k)\right] dx \\
&= 2^{-n-1/2}\sum_k h_k \left[\int_{-\infty}^{\infty} (x+k)^n \phi(x)\right] dx \\
&= 2^{-n-1/2}\sum_{i=0}^{n} \binom{n}{i} \sum_k h_k k^i \left[\int_{-\infty}^{\infty} x^{n-i}\phi(x)\right] dx. \\
&= 2^{-n-1/2}\sum_{i=0}^{n} \binom{n}{i} m_i \mu_0^{n-i},
\end{aligned} \tag{2.52}$$

where $m_i = \sum_k h_k k^i$, the ith moment of the filter coefficients h_k. Writing down the first term of the right-hand side of (2.52), we obtain

$$\mu_0^n = 2^{-n-1/2} m_0 \mu_0^n + 2^{-n-1/2}\sum_{i=1}^{n} \binom{n}{i} m_i \mu_0^{n-i},$$

and solving for μ_0^n, using the fact that $m_0 = \sum_k h_k = \sqrt{2}$, we obtain the recursive formula for the moments μ_0^n:

$$\mu_0^n = \frac{1}{(2^n-1)\sqrt{2}}\sum_{i=1}^{n} \binom{n}{i} m_i \mu_0^{n-i}, \qquad n \geq 1, \tag{2.53}$$

with $\mu_0^0 = 1$.

In the next two subsections, the moments μ_j^k are used to approximate the scaling coefficients $s_{j,k}$.

2.5.2 One-Point Quadratures

Here, we approximate $s_{j,k}$ using a one-point quadrature. Since $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$, equation (2.48) becomes

$$s_{j,k} = 2^{j/2} \int_{-\infty}^{\infty} f(x)\phi(2^j x - k)dx.$$

With the change of variable $x \to 2^j x - k$, we get

$$s_{j,k} = 2^{-j/2} \int_{-\infty}^{\infty} f(2^{-j}(x+k))\phi(x)dx. \tag{2.54}$$

Expanding $f(x)$ into a Taylor series around $2^{-j}k$,

$$f(x) = f(2^{-j}k) + \sum_{n=1}^{\infty} \frac{f^{(n)}(2^{-j}k)}{n!}(x - 2^{-j}k)^n,$$

and substituting in (2.54), using the fact that $\int_{-\infty}^{\infty} \phi(x)dx = 1$, we get

$$s_{j,k} = 2^{-j/2} f(2^{-j}k) + 2^{-j/2} \sum_{n=1}^{\infty} \frac{f^{(n)}(2^{-j}k)}{n!} 2^{-jn} \int_{-\infty}^{\infty} x^n \phi(x)dx \tag{2.55}$$

If we retain only to the first term of (2.55), we obtain a one-point quadrature approximation formula of $s_{j,k}$, i.e.,

$$s_{j,k} \approx 2^{-j/2} f(2^{-j}k). \tag{2.56}$$

which has an approximation order of $O(2^{-3j/2})$.

For more accurate approximation, we can retain higher order term and truncate the series in (2.55) at some level, say m, to obtain the approximation

$$s_{j,k} \approx \sum_{n=0}^{m} \left[\frac{2^{-j(n+1/2)}}{n!} f^{(n)}(2^{-j}k)\mu_0^n \right], \tag{2.57}$$

where μ_0^n is the nth moment of the scaling function $\phi(x)$. The approximation order of

(2.57) is $O(2^{-j(m+3/2)})$.

The quadrature (2.57) is more accurate than (2.56), however, it has the inconvenience that we have to evaluate the derivative of f as well upto m, in addition to the computation of the nth moments, μ_0^n, of the scaling function $\phi(x)$. In many application, where only samples of the function $f(x)$ are known, quadrature (2.57) would not be of any use because we do not know the samples of the derivatives of $f(x)$. This issue is considered in the next subsection, where we present a multi-point quadrature to approximate $s_{j,k}$ without using the derivatives of $f(x)$.

2.5.3 A Multi-Point Quadrature

As mentioned above, the one-point quadrature (2.57), though can be made as accurate as wanted by increasing m, it has the drawback that it needs samples of the function and of its derivatives, which may not available. To overcome this point, we present a multi-point quadrature as accurate as (2.57), which uses only samples of $f(x)$.

The multi-point quadrature approximation of the coefficients $s_{j,k}$ is given by

$$s_{j,k} \approx 2^{-j/2} \sum_{l=0}^{M} w_l f(2^{-j}(l+k)), \tag{2.58}$$

where M is a positive integer and w_l some weights to be determined such that the approximation (2.58) is exact for all polynomials of degree less or equal to M.

Let $f(x) = x^r,\ 0 \le r \le M$, be a monomial of degree $r \le M$. Then quadrature (2.58) is exact for the monomials $f(x) = x^r, 0 \le r \le M$, if and only if

$$\sum_{l=0}^{M} w_l (2^{-j}(l+k))^r = 2^{j/2} \int_{-\infty}^{\infty} x^r \phi(2^j x - k) dx, \quad 0 \le r \le M. \tag{2.59}$$

Manipulating (2.59), we obtain

$$\sum_{i=0}^{r} \left[\binom{r}{i} k^{r-i} \left(\sum_{l=0}^{M} l^i w_l \right) \right] = 2^{-j/2} \sum_{i=0}^{r} \binom{r}{i} k^{r-i} \mu_0^i, \quad 0 \le r \le M. \tag{2.60}$$

or

$$\sum_{i=0}^{r}\left[\binom{r}{i}k^{r-i}\left(\sum_{l=0}^{M}l^{i}w_{l}-2^{-j/2}\mu_{0}^{i}\right)\right]=0,\quad 0\leq r\leq M. \tag{2.61}$$

Let $\mathscr{W}_i=\sum_{l=0}^{M}l^{i}w_l$ and $\vec{C}=[C_0\,C_1\ldots C_M]^T$, $C_i=\mathscr{W}_i-2^{-j/2}\mu_0^i$. Let $\mathscr{A}$ be the $(M+1)\times(M+1)$ lower triangular matrix with the (i,j)-entry, $0\leq i,j\leq M$, given by $\mathscr{A}_{i,j}=0$ if $j>i$ and $\binom{i}{j}k^{i-j}$ if $i\geq j$. With these notations, (2.61) can be written as the homogeneous system

$$\mathscr{A}\vec{C}=\mathbf{0}. \tag{2.62}$$

Since $det(\mathscr{A})=\prod_{i=0}^{M}\mathscr{A}_{i,i}=\prod_{i=0}^{M}\binom{i}{i}=1\neq 0$, system (2.62) has the only the trivial solution, $C_i=0$ for all $0\leq i\leq M$. Thus, the weights w_l are the solutions of the system

$$\sum_{l=0}^{M}l^{i}w_{l}=2^{-j/2}\mu_{0}^{i},\qquad i=0,1,\ldots,M. \tag{2.63}$$

Let $\mathbf{w}=[w_0\,w_1\ldots w_M]^T$ and $\mathscr{M}=[\mu_0^0\,\mu_0^1\ldots\mu_0^M]^T$. Equation (2.63) is a system of linear equations for the unknown weights $\mathbf{w}$:

$$\mathbf{A}\mathbf{w}=2^{-j/2}\mathscr{M}, \tag{2.64}$$

where the matrix $\mathbf{A}$ is a nonsingular matrix given by

$$\mathbf{A}=\begin{pmatrix}1 & 1 & 1 & \cdots & 1 & 1\\ 0 & (1)^1 & (2)^1 & \cdots & (M-1)^1 & (M)^1\\ 0 & (1)^2 & (2)^2 & \cdots & (M-1)^2 & (M)^2\\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots\\ 0 & (1)^{M-1} & (2)^{M-1} & \cdots & (M-1)^{M-1} & (M)^{M-1}\\ 0 & (1)^M & (2)^M & \cdots & (M-1)^M & (M)^M\end{pmatrix}$$

with determinant

$$|\mathbf{A}| = (-1)^{\lceil M/2 \rceil} \prod_{k=1}^{M} k! \neq 0.$$

Therefore, the weights vector $\mathbf{w}$ is given by

$$\mathbf{w} = 2^{-j/2} \mathbf{A}^{-1} \mathscr{M}. \tag{2.65}$$

Chapter 3: Wavelet Time Integration Schemes

In this Chapter, we show how Daubechies' orthonormal basis are used to represent the solution of the initial value problem

$$\begin{cases} y'(t) = f(t,y), & t \in [t_0, t_f], \\ y(t_0) = y_0. \end{cases} \tag{3.1}$$

The function f on the right hand side of (3.1) is assumed to be many times differentiable. We represent (approximate) the solution $y(t)$ of (3.1) by its orthogonal projection onto a wavelet approximation space V_m. The scaling coefficients of the solution y are determined using Galerkin method. In section 3.1, we describe the *standard* wavelet Galerkin method (WGM) to determine the scaling coefficients of $y(t)$. This WGM will prove to be unstable for any wavelet order p. In section 3.2, we present a modified stable version of the WGM, which we call MWGMp. In section 3.3, we describe three ways of approximating the scaling coefficients of the right hand side function f, giving rise to three different nonlinear implicit multistep schemes for the scaling coefficients of the solution $y(t)$.

3.1 The Wavelet Galerkin Method (WGM)

In section the general wavelet Galerkin method is described to solve the general first-order initial-value problem (3.1).

As mentioned before, we use Daubechies' orthonormal scaling functions $\phi_{m,k}(t)$ as the basis functions. Let V_m be the m^{th} level approximation space and $\phi_{m,k}(t)$ be its orthonormal bases defined by

$$\phi_{m,k} = 2^{m/2}\phi(2^m t - k), \qquad k \in Z, \tag{3.2}$$

where $\phi(t)$ is the Daubechies' mother scaling function supported in the interval $[0, N-1]$, where $N = 2p$ and p is the number of vanishing moments of the corresponding

wavelet $\psi(t)$.

We approximate the solution of (3.1) by its orthogonal projection onto the approximation space V_m:

$$y(t) \approx (P_m y)(t) = \sum_{k \in Z} \tilde{c}_k \phi_{m,k}(t) \tag{3.3}$$

and determine the expansion coefficients $\tilde{c}_k$, using Galerkin approach. Since we are solving (3.1) for $t_0 \le t \le t_f$ and $\phi_{m,k}(t)$ are compactly supported, we will need a finite number of coefficients $\tilde{c}_k$ in (3.3). Precisely, we only need the coefficients $\tilde{c}_k$ where the support of $\phi_{m,k}$ intersects (non-trivially) the interval $[t_0, t_f]$. Since ϕ is supported in $[0, N-1]$, the support of $\phi_{m,k}(t)$ is $[\frac{k}{2^m}, \frac{k+N-1}{2^m}]$. Thus, the range of k is

$$2^m t_0 - N + 1 < k \qquad \text{and} \qquad k < 2^m t_f,$$

or

$$\lfloor 2^m t_0 \rfloor - N + 2 \le k \le \lceil 2^m t_f \rceil - 1.$$

To simplify notations, we can assume, without loss of generality, that $2^m t_0 = k_0$, an interger. Then in (3.3) the summation is over $k_0 - N + 2 \le k \le k_1 - 1$, where $k_1 = \lceil 2^m t_f \rceil$. The approximation (3.3) becomes

$$y(t) \approx \sum_{k=k_0-N+2}^{k_1-1} \tilde{c}_k \phi_{m,k} = \sum_{k=k_0-N+2}^{k_1-1} 2^{m/2} \tilde{c}_k \phi(2^m t - k). \tag{3.4}$$

Substituting (3.4) into (3.1), we get

$$\sum_{k=k_0-N+2}^{k_1-1} 2^{m/2} \tilde{c}_k 2^m \phi'(2^m t - k) = f(t, y(t)) + \mathscr{R}(t), \tag{3.5}$$

where $\mathscr{R}(t)$ is the residual. The expansion coefficients $\tilde{c}_k$ are determined by requiring the residual $\mathscr{R}(t)$ to be orthogonal to all *regular* scaling functions $\phi_{m,j}, j = k_0, \ldots, k_1 -$

1. We refer to $\phi_{m,j}(t)$ as regular if it is supported to the right of t_0. Thus, we have

$$\langle \mathscr{R}(t), \phi_{m,j}(t) \rangle = 0, \qquad j = k_0, \ldots, k_1 - 1, \tag{3.6}$$

where the inner product $\langle \cdot, \cdot \rangle$ is defined as follows. For any two functions f and g,

$$\langle f, g \rangle = \int_{-\infty}^{+\infty} f(t) g(t) dt. \tag{3.7}$$

We make further simplification by scaling the time variable t and letting $\tau = 2^m t$. Denote $Y(\tau) = y(t)$, $F(\tau, Y(\tau)) = f(t, y(t))$, $c_k = 2^{m/2} \tilde{c}_k$, and $\mathscr{E}(\tau) = \mathscr{R}(t)$. Then equation (3.4) becomes

$$Y(\tau) = \sum_{k=k_0-N+2}^{k_1-1} c_k \phi(\tau - k) \tag{3.8}$$

and (3.5) becomes

$$2^m \sum_{k=k_0-N+2}^{k_1-1} c_k \, \phi'(\tau - k) = F(\tau, Y(\tau)) + \mathscr{E}(\tau). \tag{3.9}$$

As mentioned above, to find the coefficients $\tilde{c}_k, k_0 - N + 2 \le k \le k_1 - 1$, we require the error $\mathscr{R}(t)$ in (3.5) to be orthogonal to all *regular* scaling functions $\phi_{m,j}, j = k_0, \ldots, k_1 - 1$. This equivalent to finding $c_k = 2^m \tilde{c}_k$ by requiring $\mathscr{E}(\tau)$ in (3.9) to be orthogonal to all $\phi(\tau - j),\ k_0 \le j \le k_1 - 1$, i.e.,

$$\langle \mathscr{E}(\tau), \phi(\tau - j) \rangle = 0, \qquad k_0 \le j \le k_1 - 1. \tag{3.10}$$

Substituting $\mathscr{E}(\tau)$ from (3.9) into (3.10), we obtain, for $j = k_0, \ldots, k_1 - 1$,

$$\sum_{k=k_0-N+2}^{k_1-1} c_k \langle \phi'(\tau - k), \phi(\tau - j) \rangle = 2^{-m} \langle F(\tau, Y(\tau)), \phi(\tau - j) \rangle. \tag{3.11}$$

The inner products $\langle \phi'(\tau-k), \phi(\tau-j)\rangle$ are given by

$$\begin{aligned}\langle \phi'(\tau-k), \phi(\tau-j)\rangle &= \int_{-\infty}^{\infty} \phi'(\tau-k)\phi(\tau-j)d\tau \\ &= \int_{-\infty}^{\infty} \phi'(\tau)\phi(\tau-(j-k))d\tau \\ &= \langle \phi'(\tau), \phi(\tau-(j-k)\rangle \equiv \Omega_{j-k},\end{aligned}$$

where $\Omega_k = \langle \phi'(x), \phi(x-k)\rangle$ are known as the *wavelet connection coefficients*. Since ϕ is supported in $[0, N-1]$, it is easy to see that $\Omega_k = 0$ for $|k| > N-2$. It is also known that $\Omega_{-k} = -\Omega_k$, hence $\Omega_0 = 0$. Before, we proceed further, we give the following proposition cencerning the calculation of Ω_k whose proof can be found in [29].

Proposition 3.1.1. *The coefficients Ω_k satisfy the following system of linear equations:*

$$\Omega_k = 2\Omega_{2k} + \sum_{i=1}^{N/2} a_{2i-1}(\Omega_{2k-2i+1} + \Omega_{2k+2i-1}) \tag{3.12}$$

with

$$\sum_k k\Omega_k = -1, \tag{3.13}$$

where the coefficients a_n (known as the autocorrelation of the filter coefficients h_k) are given by

$$a_n = 2\sum_{i=0}^{N-1-n} h_i h_{i+n}, \qquad n = 1, 2, \ldots, N-1. \tag{3.14}$$

To use (3.12) and (3.13) to calculate the Ω_k, we consider (3.12) only for $1 \leq k \leq N-2$, noting that $\Omega_{-k} = -\Omega_{-k}$ for $|k| > N-2$ and $\Omega_0 = 0$, and rewrite it as an eigenvalue problem

$$B\vec{\Omega} = \vec{\Omega}, \qquad \vec{\Omega} = (\Omega_1\ \Omega_2\ \cdots\ \Omega_{N-2})^T, \tag{3.15}$$

where the matrix B is $(N-2)\times(N-2)$ and its entries are obtained by writing down (3.12) for every $1 \leq k \leq N-2$, and collecting the coefficients of Ω_k. It follows that $\vec{\Omega}$

is the normalized eigenvector of B corresponding to the eigenvalue $\lambda = 1$.

Now back to (3.11). Let $c_j^F = \langle F(\tau, Y(\tau)), \phi(\tau - j)\rangle,\ j = k_0, \ldots, k_1 - 1$, be the scaling coefficients of $F(\tau, Y(\tau))$. Then equation (3.11) can be written as the following rectangular system of $(k_1 - k_0)$ equations in the $(k_1 - k_0 + N - 2)$ unknowns c_k:

$$\sum_{k=k_0-N+2}^{k_1-1} \Omega_{j-k}\, c_k = 2^{-m} c_j^F, \qquad j = k_0, \ldots, k_1 - 1, \tag{3.16}$$

or in matrix form

$$A\vec{c} = \vec{c}^F, \tag{3.17}$$

where $\vec{c} = (c_{k_0-N+2}\ c_{k_0-N+3}\ \cdots\ c_{k_1-1})^T$, $\vec{c}^F = (c_{k_0}^F\ c_{k_0+1}^F \cdots\ c_{k_1-1}^F)^T$, and A the rectangular matrix of size $(k_1 - k_0) \times (k_1 - k_0 + N - 2)$ with entries $A_{j,k} = \Omega_{j-k}$. It can be seen that few problems arise if (3.16) (or (3.17)) is to be used to determine the coefficients c_k. First, the matrix A is not square. More important is that $\vec{c}^F$ is not known because they involve the unknown solution $Y(\tau)$ and hence c_j. Therefore, as is, (3.16) cannot be used. Nevertheless, it will be our starting point to construct schemes which can be used.

Since $\Omega_{j-k} = 0$ for $|j - k| > N - 2$, the sum in (3.16) runs over k such that $|j - k| \le N - 2$ or $j - N + 2 \le k \le j + N - 2$. Equation (3.16) then becomes

$$\sum_{k=j-N+2}^{j+N-2} \Omega_{j-k}\, c_k = 2^{-m} c_j^F, \qquad j = k_0, \ldots, k_1 - 1, \tag{3.18}$$

which can be regarded as a multistep method to solve for the coefficients c_{j+N-2} for $k_0 \le j \le k_1 - 1$, via

$$c_{j+N-2} = -\sum_{k=j-N+2}^{j+N-3} \frac{\Omega_{j-k}}{\Omega_{-N+2}} c_k + \frac{2^{-m}}{\Omega_{-N+2}} c_j^F, \tag{3.19}$$

provided that the initial $(2N - 4)$ coefficients, c_j, for $k_0 - N + 2 \le j \le k_0 + N - 3$, and c_j^F for $j = k_0, \ldots k_1 - 1$, are known. We will see in later sections how to approximate these coefficients. We refer to scheme (3.19) as the wavelet Galerkin method (WGM).

Before we leave this section, we would like to mention that, eventhough (3.19) can be used, it is is unstable. This can be seen by considering its characteristic polynomial

$$P(\lambda) = \sum_{i=0}^{2N-4} \Omega_{N-2-i} \lambda^i, \tag{3.20}$$

which turns out that, for all Daubechies' wavelets with $N \geq 3$, it has roots outside the unit circle.

To overcome the instability drawback of the presented wavelet Galerkin method (3.19), we will see in the next section how it can be modified to obtain a stable scheme. This new scheme will be called the modified wavelet Galerkin method of order p (MWGMp), where p is the order of the Daubechies' wavelet used.

3.2 The Modified Wavelet Galerkin Method of Order p (MWGMp)

In this section, we will modify the wavelet Galerkin method presented in the previous section to obtain a stable implicit multistep scheme.

Our starting point is (3.18), namely,

$$\sum_{k=j-N+2}^{j+N-2} \Omega_{j-k} c_k = 2^{-m} c_j^F, \qquad j = k_0, \ldots, k_1 - 1. \tag{3.21}$$

Suppose that we want to solve for a particular c_j for some $k_0 \leq j \leq k_1 - 1$. This requires the knowledge of $c_{j-N+2}, \ldots, c_{j-1}$ and $c_{j+1}, \ldots, c_{j+N-2}$, in addition to c_j^F. For now suppose that we know c_j^F. The idea here is to approximate $c_{j+1}, \ldots, c_{j+N-2}$ in terms of previous c_{j-k} for some range of $k \geq 0$. This will transform (3.21) into a multistep method for c_j in terms of c_{j-k}. Since j starts from k_0, in order to use (3.21), we need to have available c_{k_0-k} for $k = 1, \ldots, N-2$, as initial values. The derivation of the MWGMp proceeds as follows.

1. Approminate $c_{j+1}, \ldots, c_{j+N-2}$ by polynomial extrapolation in terms of c_{j-k} for $k = 0, \ldots, p-1$, where p is the order of the wavelet used.

2. Approximate $c_{k_0-N+2},\ldots,c_{k_0-1}$ needed to start (3.21).

The above two approximation steps rely on the nice property of Daubechies' wavelet of order p that any polynomial of degree $p-1$ is exactly generated by integral translates of ϕ.

3.2.1 Approximation of $c_{j+1},\ldots,c_{j+N-2}$.

In this subsection, we show how to approximate $c_{j+1},\ldots,c_{j+N-2}$ by expressing them in terms of $c_{j-k},\ k=0,\ldots,p-1$.

Approximate the solution

$$Y(\tau)=\sum_{k=k_0-N+2}^{k_1-1} c_k\,\phi(\tau-k)$$

by a polynomial expansion of degree $p-1$, $P_Y(\tau)$, around $\tau=j+1$:

$$Y(\tau)\approx P_Y(\tau)=\sum_{l=0}^{p-1}\lambda_l(\tau-j-1)^l, \tag{3.22}$$

with coefficients λ_l, $l=0,1,\ldots,p-1$, to be determined.

Since any polynomial of degree less than or equal to $p-1$ is exactly generated by integral translates of $\phi(\tau)$, we can write

$$P_Y(\tau)=\sum_k c_k^{P_Y}\phi(\tau-k), \tag{3.23}$$

where $c_k^{P_Y}$ are the scaling coefficients of $P_Y(\tau)$ and are given by

$$\begin{aligned} c_k^{P_Y}=\langle P_Y(\tau),\phi(\tau-k)\rangle &= \sum_{l=0}^{p-1}\lambda_l\langle(\tau-j-1)^l,\phi(\tau-k)\rangle \\ &= \sum_{l=0}^{p-1}\lambda_l\langle\tau^l,\phi(\tau-(k-j-1))\rangle \\ &= \sum_{l=0}^{p-1}\lambda_l\mu_{k-j-1}^l, \end{aligned} \tag{3.24}$$

where μ_{k-j-1}^{l} is the lth moment of the scaling function $\phi(\tau-(k-j-1))$, discussed in Chapter 2, which can be computed exactly using equations (2.50) and (2.53).

Now, we use (3.24) to find the polynomial coefficients $\lambda_l,\ l=0,1,\ldots,p-1$, by requiring that $c_k^{P_Y}=c_k$ for $j-p+1\le k\le j$. This gives the following system

$$\underbrace{\begin{pmatrix} c_{j-p+1}\\ c_{j-p+2}\\ \vdots\\ c_{j-1}\\ c_j \end{pmatrix}}_{C_1} = \underbrace{\begin{pmatrix} \mu_{-p}^0 & \mu_{-p}^1 & \cdots & \mu_{-p}^{p-1}\\ \mu_{-p+1}^0 & \mu_{-p+1}^1 & \cdots & \mu_{-p+1}^{p-1}\\ \vdots & \vdots & \ddots & \vdots\\ \mu_0^0 & \mu_0^1 & \cdots & \mu_0^{p-1}\\ \mu_{-1}^0 & \mu_{-1}^1 & \cdots & \mu_{-1}^{p-1} \end{pmatrix}}_{M_1} \underbrace{\begin{pmatrix} \lambda_0\\ \lambda_1\\ \cdots\\ \lambda_{p-2}\\ \lambda_{p-1} \end{pmatrix}}_{\Lambda} \tag{3.25}$$

from which we solve for Λ to obtain

$$\Lambda = M_1^{-1}C_1. \tag{3.26}$$

Then using the solution for Λ in (3.26), we approximate $c_{j+1},\ldots,c_{j+N-2}$ by $c_{j+1}^{P_Y},\ldots,$ $c_{j+N-2}^{P_Y}$. From (3.24), we obtain

$$\underbrace{\begin{pmatrix} c_{j+1}\\ c_{j+2}\\ \vdots\\ c_{j+N-3}\\ c_{j+N-2} \end{pmatrix}}_{C_2} \approx \begin{pmatrix} c_{j+1}^{P_Y}\\ c_{j+2}^{P_Y}\\ \vdots\\ c_{j+N-3}^{P_Y}\\ c_{j+N-2}^{P_Y} \end{pmatrix} = \underbrace{\begin{pmatrix} \mu_0^0 & \mu_0^1 & \cdots & \mu_0^{p-1}\\ \mu_1^0 & \mu_1^1 & \cdots & \mu_1^{p-1}\\ \vdots & \vdots & \ddots & \vdots\\ \mu_{N-4}^0 & \mu_{N-4}^1 & \cdots & \mu_{N-4}^{p-1}\\ \mu_{N-3}^0 & \mu_{N-3}^1 & \cdots & \mu_{N-3}^{p-1} \end{pmatrix}}_{M_2} \underbrace{\begin{pmatrix} \lambda_0\\ \lambda_1\\ \vdots\\ \lambda_{p-2}\\ \lambda_{p-1} \end{pmatrix}}_{\Lambda},$$

or in matrix form,

$$C_2 = M_2M_1^{-1}C_1. \tag{3.27}$$

Therefore, we have the following approximation to c_{j+k} for $k = 1, \ldots, N-2$:

$$c_{j+k} = (M_2 M_1^{-1} C_1)_k = \sum_{i=1}^{p} (M_2 M_1^{-1})_{k,i} c_{j-p+i},$$

where $(M_2 M_1^{-1})_{k,i}$ is the (k,i) entry of $M_2 M_1^{-1}$. This completes the first step. In the next subsection, we show how to approximate $c_{k_0-N+2}, \ldots, c_{k_0-1}$.

3.2.2 Initialization Procedure

In this subsection we will present an initialization procedure to approximate the coefficients $c_{k_0-N+2}, \ldots, c_{k_0-1}$, needed to start (3.21).

Approximate the solution $Y(\tau)$ by a polynomial, $Q_Y(\tau)$, of degree $(p-1)$ around $j = k_0$,

$$Y(\tau) = \sum_{k=k_0-N+2}^{k_1-1} c_k \phi(\tau - k) \approx Q_Y(\tau) = \sum_{l=0}^{p-1} \lambda_l (\tau - k_0)^l. \tag{3.28}$$

Since $Q_Y(\tau) \in span(\{\phi(\tau-k),\ k \in Z\})$ (a polynomial of degree $p-1$), the scaling coefficients of $Q_Y(\tau)$, $c_k^{Q_Y} = \langle Q_Y(\tau), \phi(\tau-k) \rangle$, are given by

$$c_k^{Q_Y} = \sum_{l=0}^{p-1} \lambda_l \mu_{k-k_0}^l. \tag{3.29}$$

We approximate c_k by $c_k^{Q_Y}$ for $k = k_0 - N + 2, \ldots, k_0 - 1$. From (3.29), we obtain

$$\underbrace{\begin{pmatrix} c_{k_0-N+2} \\ c_{k_0-N+3} \\ \vdots \\ c_{k_0-2} \\ c_{k_0-1} \end{pmatrix}}_{\vec{C}_0} = \underbrace{\begin{pmatrix} \mu_{-N+2}^0 & \mu_{-N+2}^1 & \cdots & \mu_{-N+2}^{p-1} \\ \mu_{-N+3}^0 & \mu_{-N+3}^1 & \cdots & \mu_{-N+3}^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{-2}^0 & \mu_{-2}^1 & \cdots & \mu_{-2}^{p-1} \\ \mu_{-1}^0 & \mu_{-1}^1 & \cdots & \mu_{-1}^{p-1} \end{pmatrix}}_{M_3} \underbrace{\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \cdots \\ \lambda_{p-2} \\ \lambda_{p-1} \end{pmatrix}}_{\Lambda}. \tag{3.30}$$

The coefficients $\lambda_l,\ l = 0, 1, \ldots, p-1$, are found as follows. Let $t_0,\ t_1, \ldots, t_{p-1}$, be instances of time, and suppose that we have the solution at theses instances $y_i = y(t_i),\ i = 0, 1, \ldots, p-1$. Note that $y(t_0) = y_0$ is given. We let $y(t_i) = Q_Y(\tau_i),\ \tau_i = 2^m t_i$. From

(3.28), we obtain

$$\underbrace{\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{p-2} \\ y_{p-1} \end{pmatrix}}_{\vec{Y}_0} = \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & (2^m t_1 - k_0) & \cdots & (2^m t_1 - k_0)^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (2^m t_{p-2} - k_0) & \cdots & (2^m t_{p-2} - k_0)^{p-1} \\ 1 & (2^m t_{p-1} - k_0) & \cdots & (2^m t_{p-1} - k_0)^{p-1} \end{pmatrix}}_{M_4} \underbrace{\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \cdots \\ \lambda_{p-2} \\ \lambda_{p-1} \end{pmatrix}}_{\Lambda} \tag{3.31}$$

If we let $t_i = t_0 + ih$, where $h = 2^{-m}$ is the time step size, then $2^m t_i - k_0 = i$, and the matrix M_4 reduces to

$$M_4 = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2^2 & \cdots & 2^{p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (p-2) & (p-2)^2 & \cdots & (p-2)^{p-1} \\ 1 & (p-1) & (p-1)^2 & \cdots & (p-1)^{p-1} \end{pmatrix}. \tag{3.32}$$

From Eq. (3.31), we solve for Λ,

$$\Lambda = M_4^{-1} \vec{Y}_0. \tag{3.33}$$

Then from (3.30), $c_{k_0-N+2}, \ldots, c_{k_0-1}$ are given by

$$\vec{C}_0 = M_3 M_4^{-1} \vec{Y}_0. \tag{3.34}$$

We note that $y_i = y(t_i),\ i = 1, \ldots, p-1$, can be obtained by integrating (3.1) using any other one step method such as RK4. This completes the second step in approximating c_{k_0-k} for $k = 1, \ldots, N-2$, needed to start the algorithm (3.21).

3.2.3 The MWGMp

In this Subsection, we combine the results of Sections 3.2.1 and 3.2.2 to write down the modified wavelet Galerkin method of order p (MWGMp).

Recall that our starting point is equation (3.21),

$$\sum_{k=j-N+2}^{j+N-2} \Omega_{j-k} c_k = 2^{-m} c_j^F, \qquad j = k_0, \ldots, k_1 - 1. \tag{3.35}$$

First we rewrite it as

$$\sum_{k=1}^{N-2} \Omega_k c_{j-k} + \Omega_0 c_j + \sum_{k=1}^{N-2} \Omega_{-k} c_{j+k} = 2^{-m} c_j^F, \qquad j = k_0, \ldots, k_1 - 1. \tag{3.36}$$

Substituting for c_{j+k} from (3.28), we obtain, for $j = k_0, \ldots, k_1 - 1$,

$$\begin{aligned}
&\sum_{k=1}^{N-2} \Omega_k c_{j-k} + \Omega_0 c_j + \sum_{k=1}^{N-2} \Omega_{-k} \left(\sum_{i=1}^{p} (M_2 M_1^{-1})_{k,i} c_{j-p+i} \right) = 2^{-m} c_j^F \\
\Longrightarrow\ &\sum_{k=1}^{N-2} \Omega_k c_{j-k} + \Omega_0 c_j + \sum_{i=1}^{p} \left[\sum_{s=1}^{N-2} \Omega_{-s} (M_2 M_1^{-1})_{s,i} \right] c_{j-(p-i)} = 2^{-m} c_j^F \\
\Longrightarrow\ &\sum_{k=1}^{N-2} \Omega_k c_{j-k} + \Omega_0 c_j + \sum_{k=0}^{p-1} \left[\sum_{s=1}^{N-2} \Omega_{-s} (M_2 M_1^{-1})_{s,p-k} \right] c_{j-k} = 2^{-m} c_j^F .
\end{aligned} \tag{3.37}$$

Define $\Theta = (\Theta_0\ \Theta_1\ \cdots\ \Theta_{p-1}) = (\Omega_{-1}\ \Omega_{-2}\ \cdots\ \Omega_{-(N-2)}) M_2 M_1^{-1}$. Then (3.37) becomes

$$\sum_{k=1}^{N-2} \Omega_k c_{j-k} + \Omega_0 c_j + \sum_{k=0}^{p-1} \Theta_{p-k-1} c_{j-k} = 2^{-m} c_j^F . \tag{3.38}$$

If we further define

$$\rho_i = \begin{cases} \Omega_i + \Theta_{p-i-1}, & i = 0, 1, \ldots, p-1, \\ \Omega_i, & i = p, \ldots, N-2, \end{cases} \tag{3.39}$$

we can rewrite (3.38) as

$$\sum_{k=0}^{N-2} \rho_k c_{j-k} = 2^{-m} c_j^F, \qquad j = k_0, \ldots, k_1 - 1. \tag{3.40}$$

The algorithm in (3.40) is a $(N-2)$-multistep method that can be used to calculate the coefficients c_j for $j = k_0, \ldots, k_1 - 1$, with initial starting values c_{k_0-k}, $k = 1, \ldots, N-2$, given by (3.34) as explained in subsection 3.2.2. Algorithm (3.40) is referred to as the modified wavelet Galerkin method of order p (MWGMp), where p refers to the order of the wavelet used. Before we leave this section, we should mention that (3.40) assumes knowing c_j^F, which are not available. The next section deals with three ways to approximate the scaling coefficients, c_j^F, of the right hand side function $f(t, y(t)) = F(\tau, Y(\tau))$, thereby completing the MWGMp.

3.3 The MWGMp-PY, the MWGMp-PF, and the MWGMp-MP

In this section, we present three different approaches to approximate c_j^F in terms of c_{j-k}. The first approach uses a polynomial expansion of $Y(\tau)$ and the resulting complete MWGMp will be referred to as the MWGMp-PY. The second approach uses polynomial expansion of $F(\tau, Y(\tau))$, and the resulting complete MWGMp will be referred to as the MWGMp-PF. The third approach uses the multi-point quadrature discussed in Subsection 2.5.3 (see equations (2.58) and (2.65)), and the resulting complete MWGMp will be referred to as the MWGMp-MP. The first and the second approaches start with a simple trapezoidal approximation of the integral defining c_j^F. All three approaches produce implicit multistep methods for the coefficients c_j.

Recall that c_j^F is defined by

$$c_j^F = \langle F(\tau, Y(\tau)), \phi(\tau - j) \rangle = \int_{-\infty}^{\infty} F(\tau, Y(\tau)) \phi(\tau - j) d\tau. \tag{3.41}$$

Since $\phi(\tau - j)$ is supported in $[j, j + N - 1]$, we have

$$c_j^F = \int_j^{j+N-1} F(\tau, Y(\tau)) \phi(\tau - j) d\tau. \tag{3.42}$$

Approximate the integral in (3.42) by a simple trapezoidal rule to get

$$\begin{aligned} c_j^F \approx \sum_{k=j}^{j+N-1} F(k,Y(k))\phi(k-j) &= \sum_{k=j+1}^{j+N-2} F(k,Y(k))\phi(k-j) \\ &= \sum_{k=1}^{N-2} F(j+k,Y(j+k))\phi(k), \end{aligned} \tag{3.43}$$

where we have used the fact that $\phi(0) = \phi(N-1) = 0$.

Based on (3.43), we see that we need $F(j+k,Y(j+k))$, for $k = 1,\ldots,N-2$, in which $Y(j+k)$ are the unknown terms. We will approach the problem in two ways:

(i) Approximate $Y(j+k)$ in terms of c_{j-l}. This is done in Subsection 3.3.1.

(ii) Approximate $F(j+k,Y(j+k))$, all together, in terms of c_{j-l}. This is done in Subsection 3.3.2.

3.3.1 The MWGMp-PY

In this subsection, we approximate $Y(j+k)$, $k = 1,\ldots,N-2$, using a polynomial approximation of $Y(\tau)$ of degree $p-1$ around $\tau = j+1$. Approximate $Y(\tau)$ by

$$Y(\tau) \approx P(\tau) = \sum_{l=0}^{p-1} \lambda_l (\tau - (j+1))^l, \tag{3.44}$$

where the polynomial cofficients, λ_l, are to be determined. Using (3.44), we approximate $Y(j+k) \approx P(j+k) = \sum_{l=0}^{p-1} \lambda_l (k-1)^l$, for $k = 1,\ldots,N-2$. This gives the following system of equations:

$$\begin{pmatrix} Y(j+1) \\ Y(j+2) \\ Y(j+3) \\ \vdots \\ Y(j+N-3) \\ Y(j+N-2) \end{pmatrix} \approx \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 2^1 & \cdots & 2^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (N-4)^1 & \cdots & (N-4)^{p-1} \\ 1 & (N-3)^1 & \cdots & (N-3)^{p-1} \end{pmatrix}}_{V} \underbrace{\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{p-2} \\ \lambda_{p-1} \end{pmatrix}}_{\Lambda}. \tag{3.45}$$

The coefficients λ_l are determined by imposing that the scaling coefficients, c_k, of $Y(\tau)$ to be equal to the scaling coefficients, c_k^P, of $P(\tau)$, namely, $c_k = c_k^P,\ k = j-(p-1),\ldots,j$. The scaling coefficients, c_k^P, of $P(\tau)$ are given by

$$c_k^P = \langle P(\tau), \phi(\tau - k)\rangle = \sum_{l=0}^{p-1} \lambda_l \mu_{k-j-1}^l.$$

We then have

$$\underbrace{\begin{pmatrix} c_{j-p+1} \\ c_{j-p+2} \\ \vdots \\ c_{j-1} \\ c_j \end{pmatrix}}_{C_1^Y} = \underbrace{\begin{pmatrix} \mu_{-p}^0 & \mu_{-p}^1 & \cdots & \mu_{-p}^{p-1} \\ \mu_{-p+1}^0 & \mu_{-p+1}^1 & \cdots & \mu_{-p+1}^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_0^0 & \mu_0^1 & \cdots & \mu_0^{p-1} \\ \mu_{-1}^0 & \mu_{-1}^1 & \cdots & \mu_{-1}^{p-1} \end{pmatrix}}_{M_1} \underbrace{\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \cdots \\ \lambda_{p-2} \\ \lambda_{p-1} \end{pmatrix}}_{\Lambda}, \tag{3.46}$$

from which we can solve for Λ:

$$\Lambda = M_1^{-1} C_1^Y. \tag{3.47}$$

Substituting Λ from (3.47) into (3.45), we obtain the approximation

$$\begin{pmatrix} Y(j+1) \\ Y(j+2) \\ \vdots \\ Y(j+N-3) \\ Y(j+N-2) \end{pmatrix} \approx V M_1^{-1} C_1^Y.$$

Let $\mathscr{X} = V M_1^{-1}$. Then

$$Y(j+k) = \sum_{i=0}^{p-1} \mathscr{X}_{k,p-i} c_{j-i}. \tag{3.48}$$

Finally, we obtain the approximation for c_j^F as:

$$c_j^F \approx \sum_{k=1}^{N-2} \phi(k) F\left(j+k, \sum_{i=0}^{p-1} \mathscr{X}_{k,p-i} c_{j-i}\right). \tag{3.49}$$

Therefore, the MWGMp in (3.40) becomes

$$\rho_0 c_j + \sum_{k=1}^{N-2} \rho_k c_{j-k} = 2^{-m} \sum_{k=1}^{N-2} \phi(k) F\left(j+k, \sum_{i=0}^{p-1} \mathscr{X}_{k,p-i} c_{j-i}\right). \tag{3.50}$$

Let $\sigma_k^{PY} = \phi(k),\ k = 1,\ldots,N-2$. Then equation (3.50) is rewritten as

$$\rho_0 c_j + \sum_{i=1}^{N-2} \rho_i c_{j-i} = 2^{-m} \sum_{k=1}^{N-2} \sigma_k^{PY} F\left(j+k, \sum_{i=0}^{p} \mathscr{X}_{k,p-i} c_{j-i}\right). \tag{3.51}$$

Equation (3.51) is referred to as the MWGMp-PY, the modified wavelet Galerkin method of order p and the “PY" stands for the fact that we used polynomial approximation of $Y(\tau)$. We note that scheme (3.51) is an implicit $(N-2)$-multistep scheme. The next subsection presents a different way of approximating c_j^F, thereby leading to a different implicit multistep scheme.

3.3.2 The MWGMp-PF

In this subsection, we use a different approach to approximate c_j^F by approximating all together $F(j+k,Y(j+k)),\ k = 1,\ldots,N-2$, in (3.43). We do this by approximating $F(\tau,Y(\tau))$ by a polynomial of degree $p-2$ around $\tau = j$:

$$F(\tau,Y(\tau)) \approx P^F(\tau) = \sum_{l=0}^{p-2} \lambda_l (\tau - j)^l. \tag{3.52}$$

We use (3.52) to approximate $F(j+k,Y(j+k)) \equiv F_k^j$, for $k = 1,\ldots,N-2$, by

$$F_k^j \approx P^F(j+k) = \sum_{l=0}^{p-2} \lambda_l k^l, \tag{3.53}$$

or in matrix form

$$\underbrace{\begin{pmatrix} F_1^j \\ F_2^j \\ \vdots \\ F_{N-2}^j \end{pmatrix}}_{F_2} = \underbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2^{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & N-2 & \cdots & (N-2)^{p-2} \end{pmatrix}}_{V_2} \underbrace{\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{p-2} \end{pmatrix}}_{\Lambda} \tag{3.54}$$

As before, the coefficients λ_l are determined by requiring that $F(\tau_i, Y(\tau_i)) = P^F(\tau_i)$ at $\tau_i = j-p+3, \ldots, j+1$. From (3.53), we get

$$\underbrace{\begin{pmatrix} F_{-p+3}^j \\ F_{-p+4}^j \\ \vdots \\ F_1^j \end{pmatrix}}_{F_1} = \underbrace{\begin{pmatrix} 1 & (-p+3) & \cdots & (-p+3)^{p-2} \\ 1 & (-p+4) & \cdots & (-p+4)^{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}}_{V_1} \underbrace{\begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \cdots \\ \lambda_{p-2} \end{pmatrix}}_{\Lambda}, \tag{3.55}$$

from which we solve for Λ:

$$\Lambda = V_1^{-1} F_1. \tag{3.56}$$

Substituting Λ from (3.56) into (3.54), we obtain the approximation for $F_k^j = F(j+k, Y(j+k))$, $k = 1, \ldots, N-2$,

$$F_2 = V_2 V_1^{-1} F_1. \tag{3.57}$$

Define the $(N-2) \times (p-1)$ matrix $\mathscr{Y} = V_2 V_1^{-1}$. Then we have

$$\begin{aligned} F(j+k, Y(j+k)) &\approx \sum_{i=1}^{p-1} \mathscr{Y}_{k,i} F(j+i-p+2, Y(j+i-p+2)) \\ &= \sum_{i=0}^{p-2} \mathscr{Y}_{k,p-i-1} F(j+1-i, Y(j+1-i)), \end{aligned}$$

where

$$\begin{aligned} Y(j+1-i) &= \sum_{k=k_0-N+2}^{k_1-1} c_k \phi(j+1-i-k) \\ &= \sum_{k=j-i-(N-3)}^{j-i} c_k \phi(j+1-i-k) \qquad (\tilde{k} = j+1-i-k) \\ &= \sum_{\tilde{k}=1}^{N-2} c_{(j-i)+1-\tilde{k}} \phi(\tilde{k}). \end{aligned}$$

The approximation to c_j^F is then

$$c_j^F \approx \sum_{k=1}^{N-2} \sum_{i=0}^{p-2} \phi(k) \mathscr{Y}_{k,p-i-1} F\left(j+1-i, \sum_{\tilde{k}=1}^{N-2} c_{(j-i)+1-\tilde{k}} \phi(\tilde{k})\right) \tag{3.58}$$

$$= \sum_{i=0}^{p-2} \underbrace{\left(\sum_{k=1}^{N-2} \phi(k) \mathscr{Y}_{k,p-i-1}\right)}_{\sigma_i^{PF}} F\left(j+1-i, \sum_{\tilde{k}=1}^{N-2} c_{(j-i)+1-\tilde{k}} \phi(\tilde{k})\right) \tag{3.59}$$

$$= \sum_{i=0}^{p-2} \sigma_i^{PF} F\left(j+1-i, \sum_{\tilde{k}=1}^{N-2} c_{(j-i)+1-\tilde{k}} \phi(\tilde{k})\right) \tag{3.60}$$

Therefore, the MWGMp in (3.40), for $j = k_0, \ldots, k_1 - 1$, becomes

$$\rho_0 c_j + \sum_{i=1}^{N-2} \rho_i c_{j-i} = 2^{-m} \sum_{i=0}^{p-2} \sigma_i^{PF} F\left(j+1-i, \sum_{k=1}^{N-2} c_{(j-i)+1-k} \phi(k)\right). \tag{3.61}$$

Scheme (3.61) is referred to as the MWGMp-PF, in reference to the fact that we have used polynomial approximation of the right hand side function $F(\tau, Y(\tau))$ in approximating c_j^F.

3.3.3 The MWGMp-MP

In this section, we present another approximation to c_j^F by considering a multipoint approximation formula that is exact for polynomials of degree less than or equal to $p-2$.

We have

$$c_j^F = \int_{-\infty}^{\infty} F(\tau, Y(\tau)) \phi(\tau - j) d\tau.$$

Approximate c_j^F by the following multi-point quadrature

$$c_j^F \approx \sum_{l=0}^{p-2} w_l F(j+1-l, Y(j+1-l)), \tag{3.62}$$

with weights w_l to be determined such that the approximation is exact when $F(\tau, Y(\tau))$ is monomial of degree $\leq p-2$, i.e.,

$$\sum_{l=0}^{p-2} w_l (j+1-l)^r = \int_{-\infty}^{\infty} \tau^r \phi(\tau - j) d\tau, \qquad 0 \leq r \leq p-2. \tag{3.63}$$

Manipulating Equation (3.63), we obtain

$$\sum_{i=0}^{r} \underbrace{j^{r-i}}_{\mathscr{A}_{r,i}} \underbrace{\sum_{l=0}^{p-2} w_l (1-l)^i}_{\mathscr{W}_i} = \sum_{i=0}^{r} \underbrace{j^{r-i}}_{\mathscr{A}_{r,i}} \underbrace{\int_{-\infty}^{\infty} \tau^i \phi(\tau) d\tau}_{\mu_0^i}, \qquad 0 \leq r \leq p-2, \tag{3.64}$$

which can be written in the following matrix-vector equation

$$\mathscr{A}\mathscr{W} = \mathscr{A}\mathscr{M}, \tag{3.65}$$

where $\mathscr{A}$ is a lower triangular matrix of size $(p-1)\times(p-1)$ whose entries are $\mathscr{A}_{r,i} = j^{r-i}$, $0 \leq r,i \leq p-2$, $\mathscr{W}$ is a $(p-1)$-column vector whose entries are $\mathscr{W}_i = \sum_{l=0}^{p-2} w_l (1-l)^i$, $i = 0,\ldots,p-2$, and $\mathscr{M}$ is a $(p-1)$-column vector whose entries are $\mathscr{M}_i = \mu_0^i$, $i = 0,\ldots,p-2$. Since the matrix $\mathscr{A}$ is non-singular, with determinant $|\mathscr{A}| = \prod_{i=0}^{p-2} \mathscr{A}_{i,i} = 1$, we conclude that $\mathscr{W}_i = \mathscr{M}_i$, $0 \leq i \leq p-2$. Therefore, the weights w_l satisfy the following system

$$\sum_{l=0}^{p-2} w_l (1-l)^i = \mu_0^i, \qquad i = 0,\ldots,p-2. \tag{3.66}$$

from which we can solve for the required weights w_l to be used in the approximation (3.62).

From (3.62), we need to know $Y(j+1-l)$, $l = 0,\ldots,p-2$. We have

$$Y(j+1-l) = \sum_{k=k_0-N+2}^{k_1-1} c_k \phi(j+1-l-k).$$

Using the fact that $\phi(l) = 0$ for $l \notin \{1,2,\ldots,N-2\}$ and using the index shifting $k \to j-l+1-k$, we have

$$Y(j-l+1) = \sum_{k=1}^{N-2} c_{j-l+1-k}\phi(k).$$

It follows that

$$c_j^F \approx \sum_{l=0}^{p-2} w_l F\left(j+1-l, \sum_{k=1}^{N-2} c_{j+1-l-k}\phi(k)\right). \tag{3.67}$$

Finally, with this multi-point approximation of c_j^F, the MWGMp in (3.40), for $j = k_0,\ldots,k_1-1$, becomes

$$\rho_0 c_j + \sum_{i=1}^{N-2} \rho_i c_{j-i} = 2^{-m} \sum_{i=0}^{p-2} \sigma_i^{MP} F\left(j+1-i, \sum_{k=1}^{N-2} c_{(j-i)+1-k}\phi(k)\right), \tag{3.68}$$

which we refer to as the MWGMp-MP, where $\sigma_i^{MP} = w_i$ and w_i are the solution of (3.66).

3.4 Summary of the Different MWGMp

In the previous section, three different wavelet-based multistep schemes were derived. In this section, we summarize the formulae of each scheme. Each of the three methods gives implicit multi-step scheme for the calculation of the scaling coefficients $\tilde{c}_k = 2^{-m/2}c_k,\ k = k_0-N+2,\ldots,k_1-1$, of the approximate solution

$$y(t) \approx \sum_{k=k_0-N+2}^{k_1-1} \tilde{c}_k \phi_{m,k}(t)$$

of the IV.P.

$$y'(t) = f(t,y), \qquad t_0 < t < t_f,$$
$$y(t_0) = y_0,$$

where $N = 2p$, p is the order of Daubechie wavelet used, and $k_0 = 2^m t_0$ and $k_1 = \lceil t_f \rceil$.

1. **The MWGMp-PY:**

$$\rho_0 c_j + \sum_{k=1}^{N-2} \rho_k c_{j-k} = h \sum_{k=1}^{N-2} \sigma_k^{PY} F\left(j+k, \sum_{i=0}^{p-1} \mathscr{X}_{k,p-i} c_{j-i} \right), \tag{3.69}$$

where $h = 2^{-m}$, $\sigma_k^{PY} = \phi(k)$, and $\mathscr{X} = VM_1^{-1}$, V and M_1 are given as in equations (3.45) and (3.46), respectively.

2. **The MWGMp-PF:**

$$\rho_0 c_j + \sum_{i=1}^{N-2} \rho_i c_{j-i} = h \sum_{i=0}^{p-2} \sigma_i^{PF} F\left(j+1-i, \sum_{k=1}^{N-2} c_{(j-i)+1-k} \phi(k) \right), \tag{3.70}$$

where $h = 2^{-m}$, $\sigma_i^{PF} = \sum_{k=1}^{N-2} \phi(k) \mathscr{Y}_{k,p-i-1}$, $\mathscr{Y} = V_2 V_1^{-1}$ with V_1 and V_2 are given as in equations (3.54) and (3.55), respectively.

3. **The MWGMp-MP:**

$$\rho_0 c_j + \sum_{i=1}^{N-2} \rho_i c_{j-i} = h \sum_{i=0}^{p-2} \sigma_i^{MP} F\left(j+1-i, \sum_{k=1}^{N-2} c_{(j-i)+1-k} \phi(k) \right), \tag{3.71}$$

where $h = 2^{-m}$, $\sigma_i^{MP} = w_i$, w_i are the solution of (3.66).

In all above three schemes ρ_i are given by (3.39).

Remark. 1. Before closing this section, we have three important remarks to mention.

1. We note that the MWGMp-PF and the MWGMp-MP are identical. It turned out that for all wavelet orders p considered, we found that $\sigma_i^{PF} = \sigma_i^{MP}$, for all $0 \le i \le p-2$. So we will consider only one of them, the MWGMp-PF.

2. For the MWGMp-PF, equation (3.70), one needs c_{j-i} for $i = 1, \ldots, 3p-5$. In order to be able use (3.70) for $j \ge k_0$, it is required that $j-(3p-5) \ge k_0 - N + 2$ or $j - k_0 \ge p-3$. Thus, p must be $p \le 3$. Therefore, if $p \ge 4$, we can't use (3.70) for $j = k_0, \ldots, k_0 + p - 4$. In this case ($p \ge 4$), we first use $(p-3)$ steps of the MWGMp-PY for $j = k_0, \ldots, k_0 + p - 4$, and continue with (3.70) for $j \ge$

$k_0 + p - 3$.

3. All three schemes are implicit. If the function F is nonlinear, we would have a nonlinear equation for c_j. So, to solve for c_j, one would need to use a nonlinear solver or use a predictor-corrector approach. We opt for the latter. The prediction formula for c_j is obtained by the same way as in subsection 3.2.1 to get

$$c_j = (\mu_0^0 \;\; \mu_0^1 \;\; \dots \;\; \mu_0^{p-1}) \underbrace{\begin{pmatrix} \mu_{-p}^0 & \mu_{-p}^1 & \cdots & \mu_{-p}^{p-1} \\ \mu_{-p+1}^0 & \mu_{-p+1}^1 & \cdots & \mu_{-p+1}^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_0^0 & \mu_0^1 & \cdots & \mu_0^{p-1} \\ \mu_{-1}^0 & \mu_{-1}^1 & \cdots & \mu_{-1}^{p-1} \end{pmatrix}^{-1}}_{M_1^{-1}} \begin{pmatrix} c_{j-p} \\ c_{j-p+1} \\ \vdots \\ c_{j-1} \end{pmatrix}. \tag{3.72}$$

Chapter 4: Order of Convergence and Stability

In this Chapter, we investigate the order of convergence and stability of the three modified wavelet schemes derived in the previous chapter. First, a brief review of some definitions and theorems concerning order of convergence and stability analysis of multistep methods is given in section 4.1. Then in sections 4.2 and 4.3, we investigate the order of convergence and stability of the different MWGMp derived in the previous Chapter.

4.1 Multistep Methods, Order of Convergence and Stability Analysis

Definition 4.1.1. A general m-step multistep method for numerically solving initial value problems

$$y' = f(t,y), \qquad t_0 < t < t_f, \;\; y(t_0) = y_0, \tag{4.1}$$

can be written in the general form

$$w_j + \sum_{k=1}^{m} a_k w_{j-k} = hF(t_j, w_j, w_{j-1}, \ldots, w_{j-m}), \qquad j \geq m, \tag{4.2}$$

where $w_j \approx y(t_j),\ t_j = t_0 + jh,\ h = \frac{t_f - t_0}{n}$, and $a_k,\ 0 \leq k \leq m-1$, are constants that define the multistep method.

The initial conditions $w_0,\ w_1, \ldots, w_{m-1}$ of (4.2) are usually obtained using one-step explicit methods, such as Runge Kutta method, except $w_0 = y_0$. The most widely used multistep step methods are the linear ones defined below.

Definition 4.1.2. A linear m-step multistep method has the form

$$w_j + \sum_{k=1}^{m} a_k w_{j-k} = h \sum_{k=0}^{m} b_k f(t_{j-k}, w_{j-k}), \qquad j \geq m. \tag{4.3}$$

Note that if $b_0 = 0$ then the method is explicit and if $b_0 \neq 0$ the method is implicit.

The following are some examples of widely used linear multistep methods:

- The trapezoid rule, a one-step implicit method $(m = 1)$:

$$w_j = w_{j-1} + \frac{h}{2}\left[f(t_{j-1}, w_{j-1}) + f(t_j, w_j)\right], \qquad j \geq 1.$$

- The $4-$step explicit Adam-Bashforth method: $(m = 4)$:

$$\begin{aligned} w_j &= w_{j-1} + \frac{h}{2}\Big[55f(t_{j-1}, w_{j-1}) - 59f(t_{j-2}, w_{j-2}) \\ &\qquad +37f(t_{j-3}, w_{j-3}) - 9f(t_{j-4}, w_{j-4})\Big], \ j \geq 4. \end{aligned}$$

- The $3-$step implicit Adam-Multoun method $(m = 3)$:

$$\begin{aligned} w_j &= w_{j-1} + \frac{h}{24}\Big[9f(t_j, w_j) + 19f(t_{j-1}, w_{j-1}) \\ &\qquad -5f(t_{j-2}, w_{j-2}) + f(t_{j-3}, w_{j-3})\Big], \ j \geq 3. \end{aligned}$$

- The general m-step explicit Adam-Bashforth method has the formula

$$w_j = w_{j-1} + h\sum_{k=0}^{m-1}(-1)^k \nabla^k f(t_{j-1}, w_{j-1}) \int_0^1 \binom{-s}{k} ds$$

- The general m-step implicit Adam-Moulton method has the formula

$$w_j = w_{j-1} + h\sum_{k=0}^{m}(-1)^k \nabla^k f(t_j, w_j) \int_0^1 \binom{s}{k} ds,$$

where $\nabla^k f(t_n, w_n)$ is defined recursively, by

$$\nabla^k f(t_n, w_n) = \nabla(\nabla^{k-1} f(t_n, w_n)), \qquad \nabla f(t_n, w_n)) = f(t_n, w_n) - f(t_{n-1}, w_{n-1})$$

and

$$\binom{r}{k} = \frac{r(r-1)(r-2)\cdots(r-k+1)}{k!}.$$

Definition 4.1.3. The local truncation error, T_j, of an m-step multistep method of the form (4.2) is defined by

$$T_j(h) = \frac{y(t_j) + \sum_{k=1}^{m} a_k y(t_{j-k})}{h} - F(t_j, y(t_j), y(t_{j-1}, \ldots, y(t_{j-m})). \tag{4.4}$$

For linear m-step multistep methods (4.3), the truncation error is given by

$$T_j(h) = \frac{y(t_j) + \sum_{k=1}^{m} a_k y(t_{j-k})}{h} - \sum_{k=0}^{m} b_k f(t_{j-k}, y(t_{j-k})). \tag{4.5}$$

Definition 4.1.4. A multistep method is order p accurate if $T_j(h) = O(h^p)$ as $h \to 0$. We say that the order of convergence is $O(h^p)$.

Now we turn to talk about stability of multistep methods. We will consider the zero-stability, absolute stability and A-stability, defined below.

Definition 4.1.5. The characteristic polynomial of an m-step multistep method of the form (4.2) is the polynomial (of degree m)

$$\mathscr{P}(\lambda) = \lambda^m - \sum_{k=1}^{m} a_k \lambda^{m-k}. \tag{4.6}$$

Definition 4.1.6. (**The root condition**) An m-step multistep method is said to satisfy the *root condition* if all roots, λ_i, of its characteristic polynomial (4.6) satisfy $|\lambda_i| \leq 1$ and all roots with modulus 1 are simple.

The zero stability of an m-step multistep method depends directly on the roots, λ_i, of its characteristic polynomial $\mathscr{P}(\lambda)$ in (4.6). We have the following stability theorem.

Theorem 4.1.1. *Am m-step multistep method is zero stable (or stable) if and only if it satisfies the root condition.*

An m-step multistep method is classified as strongly stable, weekly stable or unstable as defined below.

Definition 4.1.7. An m-step multistep method is said to be

(i) **strongly stable** if it satisfies the root condition and if there is a root with modulus 1 then that root is $\lambda = 1$.

(ii) **weekly stable** if it satisfies the root condition and have two or more distinct roots with modulus 1.

(iii) **unstable** if it does not satisfy the root condition, i.e, there exists at least one λ_i with $|\lambda_i| > 1$.

Next, we discuss absolute stability. The absolute stability of an m-step multistep method is studied by looking at the stability of the method when applied to the linear model problem

$$y'(t) = ay(t), \qquad y(t_0) = y_0, \tag{4.7}$$

where $a \in \mathbb{C}$, a constant. An m-step multistep method when applied to (4.7) reduces to the linear multistep method

$$\sum_{k=0}^{m} (a_k - ahb_k) w_{j-k} = 0, \tag{4.8}$$

whose characteristic polynomial is

$$\mathscr{P}(\lambda) = A(\lambda) - zB(\lambda) = 0, \tag{4.9}$$

where $A(\lambda) = \sum_{k=0}^{m} a_k \lambda^{m-k}$, $B(\lambda) = \sum_{k=0}^{m} b_k \lambda^{m-k}$ and $z = ah$.

Absolute stability and the so-called region of absolute stability and A-stability

of a multistep method are defined below.

Definition 4.1.8. A multistep method is called absolutely stable in an open set $\mathscr{R}_A$ of the complex plane $\mathbb{C}$ if for all $z = ah \in \mathscr{R}_A$, all roots λ_k of $\mathscr{P}(\lambda)$ in (4.9) satisfy $|\lambda_k| < 1$. The set $\mathscr{R}_A$ is called the region of absolute stability of the multistep . A multistep method is called A-stable if $\mathscr{R}_A$ contains all of the left-half plane.

A method called the Boundary Locus Method [41, 42] is one of the methods for finding $\mathscr{R}_A$. The idea of the method is to determine the set of z in the z-complex plane, call it $\mathscr{C}$, for which the characteristic equation (4.9) has a least one root with modulus 1. The z-complex plane is the union of $\mathscr{R}_A$, $\mathscr{C}$, and $\mathscr{R}_{\bar{A}}$, where $\mathscr{R}_{\bar{A}}$ is the set in the z-complex plane for which (4.9) has at least one root with modulus greater than 1. If $\mathscr{C}$ is a closed curve, then $\mathscr{R}_A$ will be the interior or the exterior of such a curve.

The $\mathscr{C}$ is determined by letting $\lambda = e^{i\theta}$, $0 \leq \theta \leq 2\pi$, and solving (4.9) for z. This gives

$$z(e^{i\theta}) = \frac{A(e^{i\theta})}{B(e^{i\theta})}. \tag{4.10}$$

The curve $\mathscr{C}$ can be seen as the mapping of the unit circle of the λ-complex plane under (4.10).

In the next two sections, we investigate the order of convergence and the stability of the three modified wavelet Galerkin methods developed in the previous chapter.

4.2 Order of Convergence of the MWGMp

According to equation (4.4), the order of convergence of a multistep method is determined by its local truncation error. However, for the three different MWGMp developed in the previous chapter, we will not use (4.4) to deduce their order of convergence. Since their derivation was based on using the approximation property of Daubechies' scaling function $\phi(x)$ of order p, in that all polynomials polynomials of degree $p-1$ are exactly generated by linear combination of integral translates of $\phi(x)$,

we used this fact.

Suppose that the exact solution of the initial value problem $y'(t) = f(t,y)$, $y(t_0) = y_0$, is a polynomial of degree less than or equal to $p-1$. Then the orthogonal projection $(P_m y)(t) = y(t)$, and hence approximating the exact solution by $(P_m y)(t)$ does not produce any truncation error. We remark that in the remaining derivations as described in the previous chapter, we always use approximation by polynomials of degree $p-1$, hence there are no truncation errors produced. Therefore, we can deduce that the local truncation error of all three methods is $O(h^p)$, where $h = 2^{-m}$ is the step size.

4.3 Stability Analysis of the MWGMp

In this section, we investigate the zero stability and the absoilute stability of the MWGMp-PY and the MWGMp-PF.

The zero stability is determined by the roots of the characteristic polynomial of the method as $h \to 0$. Since the methods (MWGMp-PY and MWGMp-PF) have the same form, that is,

$$\sum_{i=0}^{N-2} \rho_i c_{j-i} = hF(t_j, c_j, c_{j-1}, \ldots, c_{j-q}), \tag{4.11}$$

where $q = 2(p-1)$ for MWGMp-PY and $q = (3p-5)$ for MWGMp-PF, they have the same characteristic polynomial (as $h \to 0$)

$$\mathscr{P}(\lambda) = \sum_{k=0}^{N-2} \rho_k \lambda^{N-2-k}. \tag{4.12}$$

Numerical experiments, reveal that for all considered Daubechies' wavelets with order $2 \leq p \leq 10$, $\mathscr{P}(\lambda)$ in (4.12) has $\lambda = 1$ as a simple root and the rest of the roots are inside the unit circle. Therefore, all three MWGMp are all strongly stable.

For the absolute stability, as described in the previous section, we need to investigate the characteristic polynomial of the methods as they are applied to the linear

model problem (4.7).

For the MWGMp-PY when applied to the linear problem (4.7), it reduces to the linear multistep method:

$$\sum_{i=0}^{N-2} \rho_i c_{j-i} = ah \sum_{i=0}^{p-1} \left(\sum_{k=1}^{N-2} \sigma_k^{PY} \mathscr{X}_{k,p-i} \right) c_{j-i}, \tag{4.13}$$

which can be written in the form

$$\sum_{i=0}^{N-2} (\rho_i - ah\sigma_i) c_{j-i} = 0, \tag{4.14}$$

where ρ_i as defined before in (3.39) and $\sigma_i = \sum_{k=1}^{N-2} (\sigma_k^{PY} \mathscr{X}_{k,p-i})$ if $0 \le i \le p-1$, and 0, otherwise. In fact it can be verified (at least numerically) that $\sigma_i = 1$ if $i = 0$, and 0, otherwise.

The characteristic polynomial of (4.14) is

$$\mathscr{P}(\lambda) = A(\lambda) - zB(\lambda), \qquad z = ah, \tag{4.15}$$

where $A(\lambda) = \sum_{i=0}^{N-2} \rho_i \lambda^{N-2-i}$ and $B(\lambda) = \sum_{i=0}^{N-2} \sigma_i \lambda^{N-2-i}$.

We applied the Boundary Locus Method as described in the previous section, we find that the regions of absolute stability of the MWGMp-PY method for $p = 2, \ldots, 5$, are as displayed in Figure **??** (left), where the absolute stability region, $\mathscr{R}_A$, is outside the closed curves. We note that as p increases the absolute stability region shrinks. A zoom around $z = 0$ of Fig. **??** (left) is displayed in Fig. **??** (right) which shows that only for $p = 2$ and $p = 3$, the method is A-stable.

The MWGMp-PF, when applied to the linear model equation, reduces to a linear multistep method of the form

$$\sum_{i=0}^{N-2} \rho_i c_{j-i} = z \sum_{i=0}^{N+p-5} \tilde{\sigma}_i c_{j-i}, \qquad z = ah, \tag{4.16}$$

where ρ_i is as defined before in (3.39). The coefficients $\tilde{\sigma}_i$, after simplification, turn

out to be given by

$$\tilde{\sigma}_i = \sum_{l=\max(0,i-N+3)}^{\min(i,p-2)} \sigma_l^{PF} \phi(i+1-l), \tag{4.17}$$

where $\sigma_l^{PF} = \sum_{k=1}^{N-2} \phi(k) \mathscr{Y}_{k,p-l-1}$, $\mathscr{Y} = V_2 V_1^{-1}$ with V_1 and V_2 are given as in equations (3.54) and (3.55), respectively.

The characteristic polynomial of (4.16) is

$$\mathscr{P}(\lambda) = \sum_{i=0}^{N+p-5} (\tilde{\rho}_i - z\tilde{\sigma}_i)\lambda^{N+p-5-i}, \tag{4.18}$$

where $\tilde{\rho}_i = \rho_i$ for $0 \leq i \leq N-2$ and 0 for $i > N-2$.

The Boundary Locus Method has been applied to find the regions of absolute stability for $p = 2,3,\ldots,7$, of the MWGMp-PF. The results are displayed in Figures 4.2- 4.4. For $p = 2,3,4$, and 5 (Figures 4.2 - 4.3), the absolute stability region is outside the closed curve. For $p = 6$ and 7 (Figure 4.4), the absolute stability region is inside the closed curve and to the left of the imaginary axis. We also notice, that only for $p = 2$ and $p = 3$, the MWGMp-PF, is A-stable.

For comparison purposes, the stability region of he Adams-Mouton q-step method for $q = 2,3,4$, and 5, are displayed in Figures 4.5 - 4.6, where it can be seen that none of these methods is A-stable, in contrast to the MWGMp-PF which is A-stable for $p = 2$ and 3. The MWGMp-PF is a 2-step method for $p = 2$ and a $(3p-5)$-step method for $p \geq 3$, with local truncation error (order of convergence) $O(h^p)$, whereas the Adams-Mouton q-step method is $O(h^{q+1})$. The Adams-Mouton q-step method has a better convergence rate, however the MWGMp-PF has a better stability property. Similarly, the MWGMp-PY has a better stability property but lower order of convergence as compared to the Adams-Mouton q-step method.

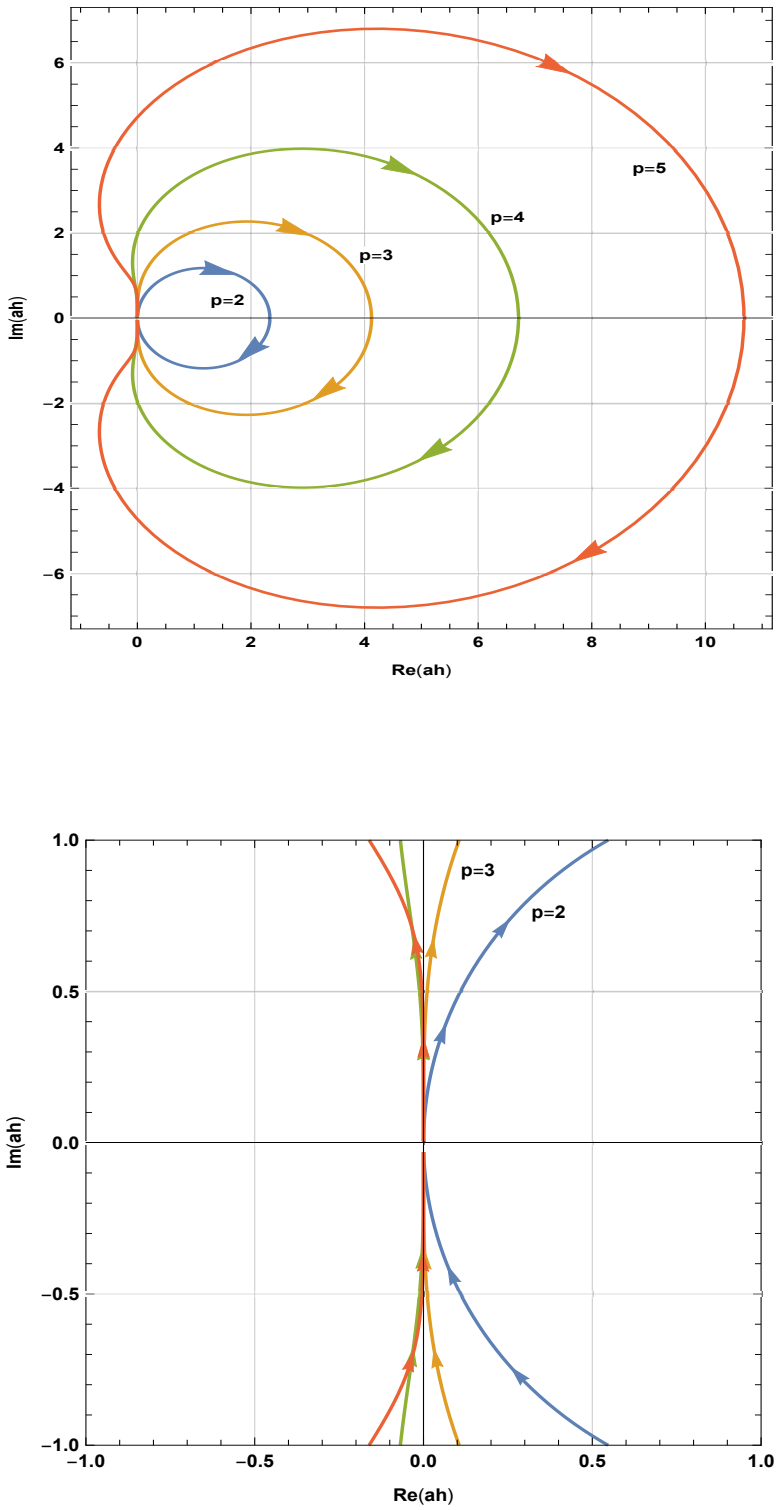


Figure 4.1: MWGMp-PY stability regions for $p = 2, \ldots, 5$

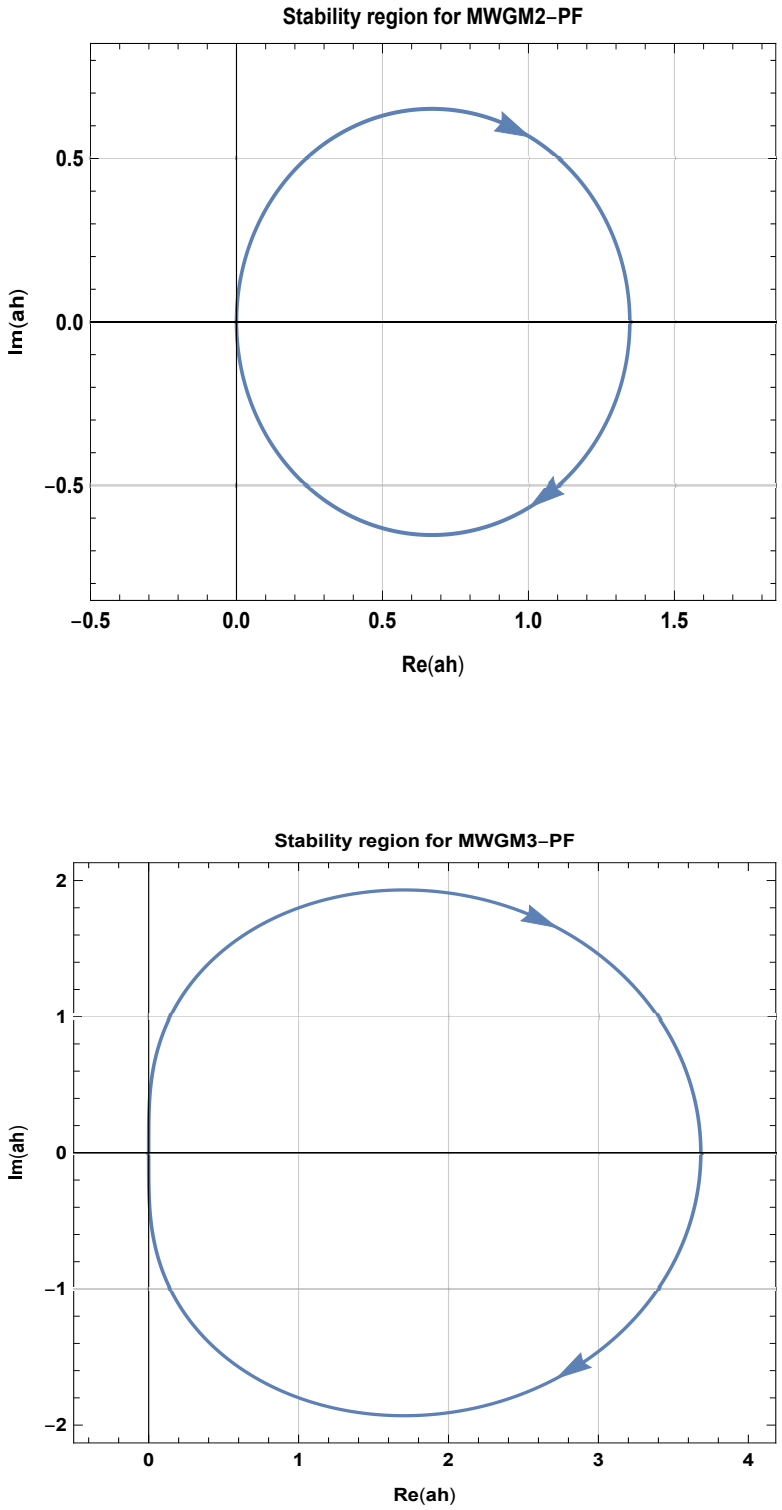


Figure 4.2: MWGMp-PF stability region, $p = 2$ (top) and $p = 3$ (bottom)

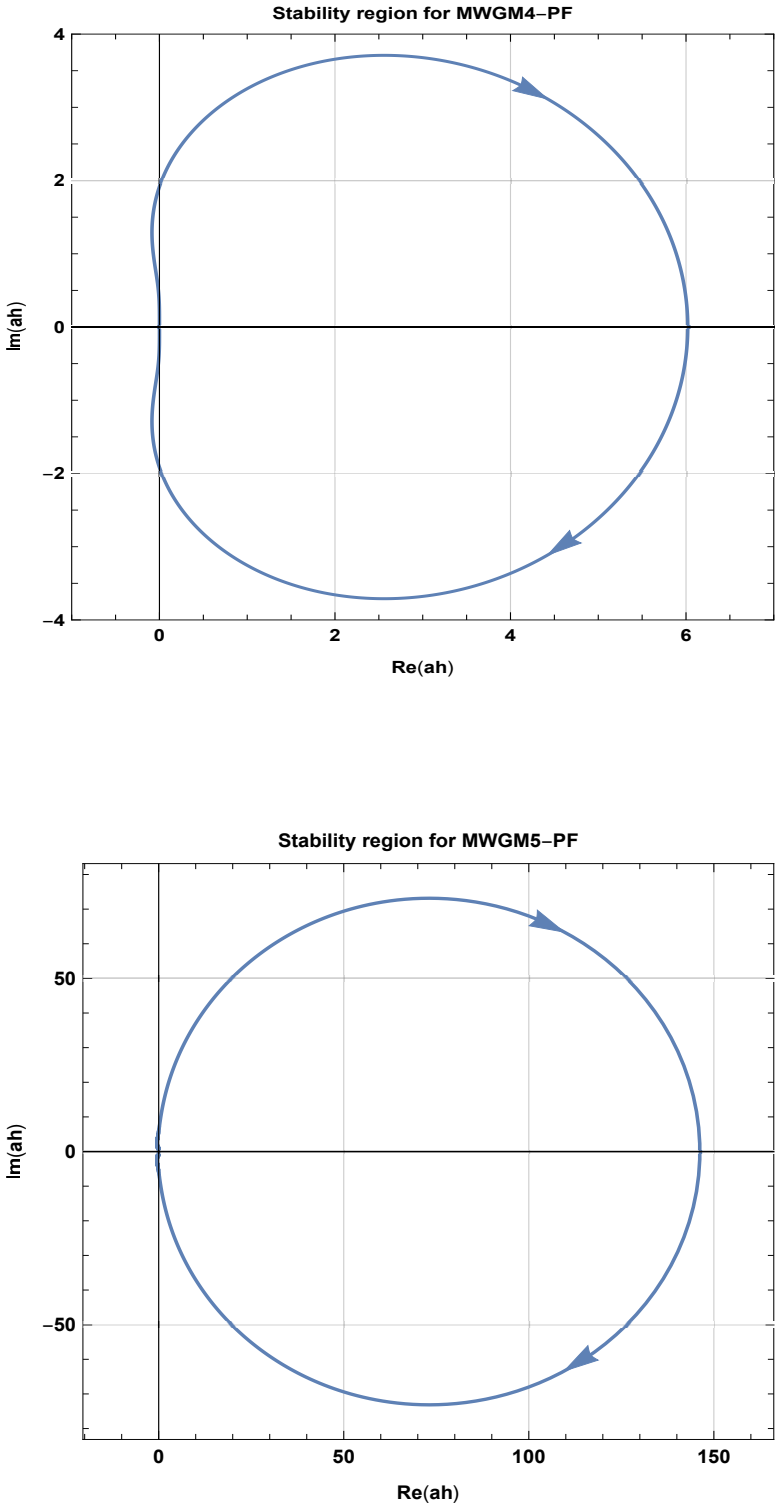


Figure 4.3: MWGMp-PF stability region, $p = 4$ (top) and $p = 5$ (bottom)

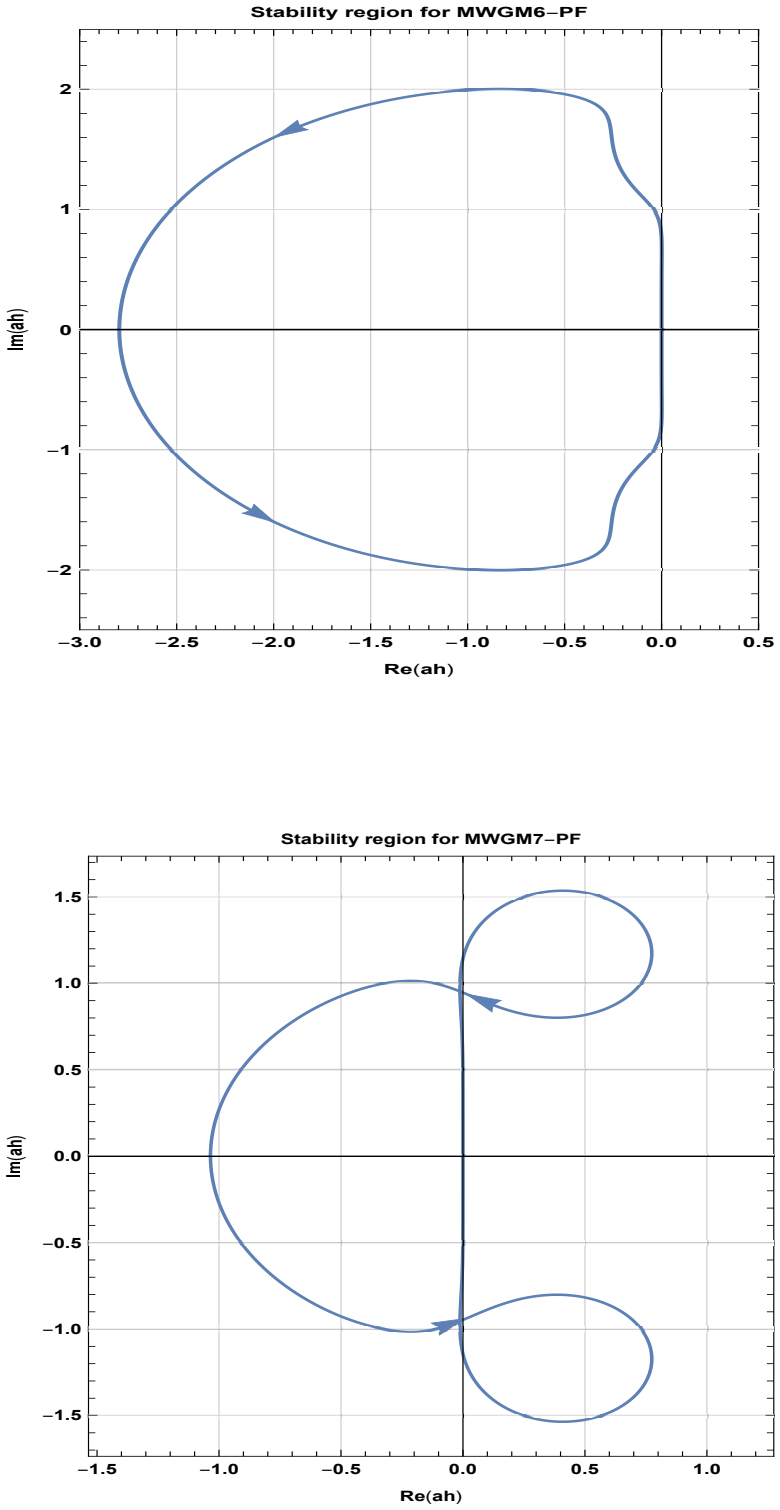


Figure 4.4: MWGMp-PF stability region, $p = 6$ (top) and $p = 7$ (bottom)

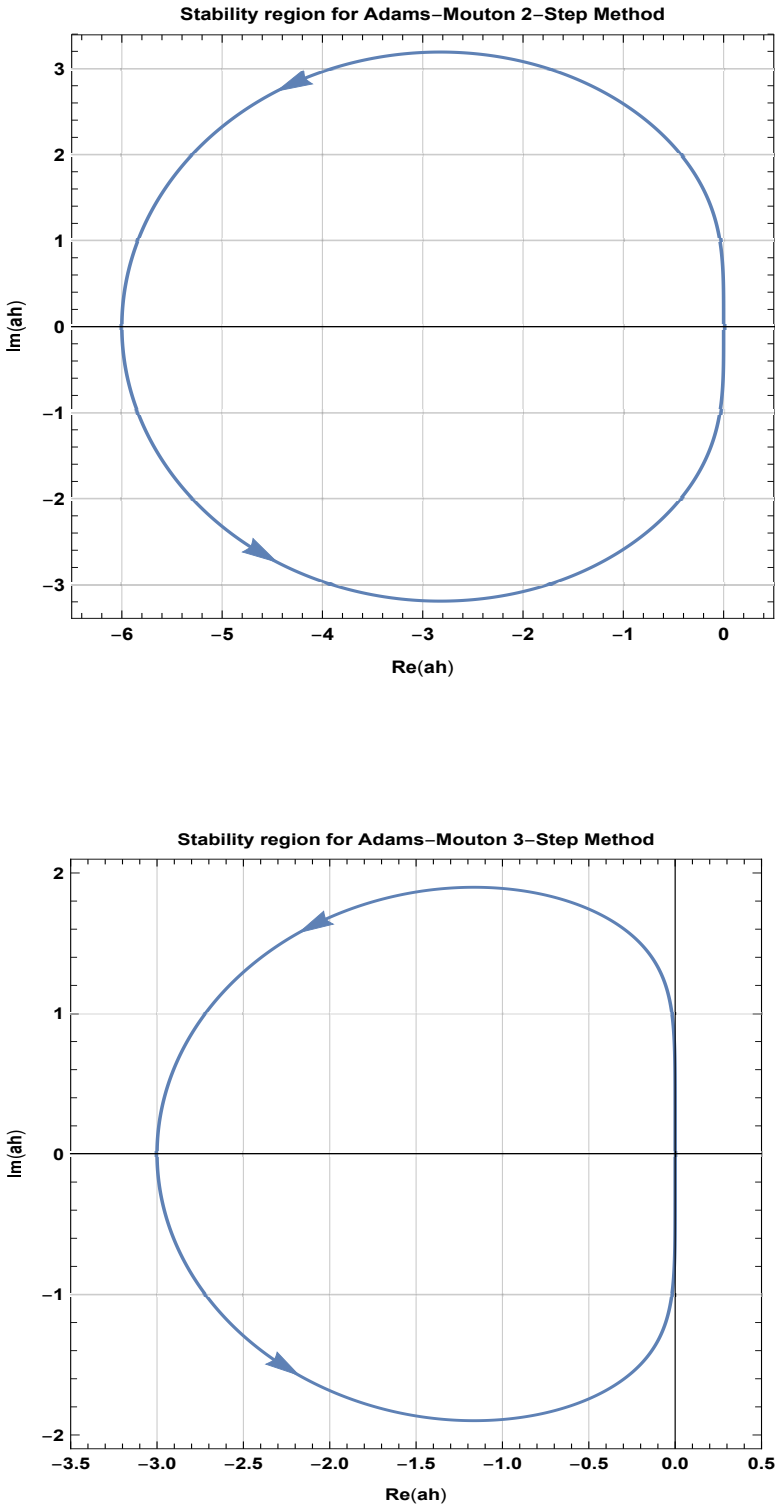


Figure 4.5: Adams-Mouton stability regions, 2-step (top) and 3-step (bottom)

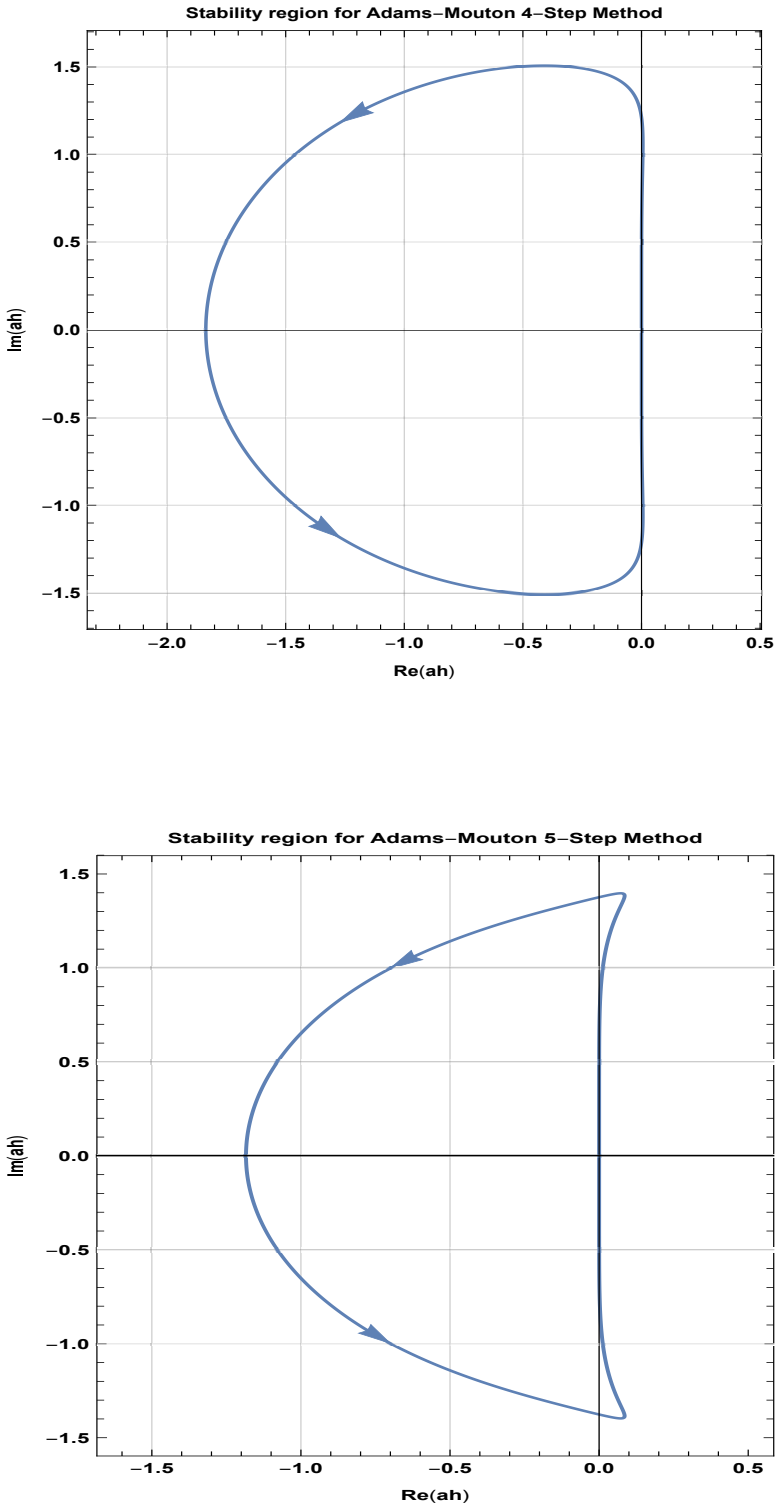


Figure 4.6: Adams-Mouton stability regions, 4-step (top) and 5-step (bottom)

Chapter 5: Numerical Examples

In this Chapter, we apply the wavelet-based multistep methods developed in the previous chapter to a number of linear and nonlinear test problems.

5.1 Introduction

Let $y(t)$ be the exact solution of the I.V.P.

$$y'(t) = f(t,y), \qquad t \in [t_0, t_f], \tag{5.1}$$

$$y(t_0) = y_0. \tag{5.2}$$

The solution $y(t)$ is first approximated by its orthogonal projection onto a wavelet space V_m with orthogonal basis $\phi_{m,k}(t) = 2^{m/2}\phi(2^m - k)$, where $\phi(t)$ is Daubechies' scaling function of order p, supported in $[0, N-1]$ with $N = 2p$, namely

$$y(t) \approx y_{m,p}(t) = (P_m y)(t) = \sum_{k=k_0-N+2}^{k_1-1} 2^{m/2}\tilde{c}_k \phi(2^m t - k). \tag{5.3}$$

We note here that if the solution $y(t)$ is a polynomial of degree $p-1$ or less then $y_{m,p}(t) \approx y(t)$. The wavelet-based multistep methods derived in the previous chapter approximate the scaling coefficients $\tilde{c}_k$ by solving for c_k ($\tilde{c}_k = 2^{-m/2}c_k$). So when these c_k are substituted into (5.3), we have the approximate solution

$$\tilde{y}_{m,p}(t) = \sum_{k=k_0-N+2}^{k_1-1} c_k \phi(2^m t - k). \tag{5.4}$$

At the discrete points $t_i = t_0 + i/2^m = 2^{-m}(k_0 + i),\ i = 0, \ldots, k_1 - k_0$, we have

$$\tilde{y}_{m,p}(t_i) = \sum_{k=k_0-N+2}^{k_1-1} c_k \phi(k_0 + i - k) = \sum_{k=1}^{N-2} c_{k-(k_0+i)} \phi(k). \tag{5.5}$$

To verify the accuracy of the methods, we use both the global error $E_g(t)$ and

its L^2-norm, $\|E_g(t)\|_2$. The global error is defined by

$$E_g(t) = |y_{exact}(t) - \tilde{y}_{m,p}(t)| \tag{5.6}$$

and its L_2-norm is defined by

$$\|E_g\|_2^2 = \int_{t_0}^{t_f} |y_{exact}(t) - \tilde{y}_{m,p}(t)|^2 dt \approx 2^{-m} \sum_i E_g^2(t_i). \tag{5.7}$$

where $y_{exact}(t)$ is the exact solution of the problem, if it is available. If the exact solution is not available, $E_g(t)$ is defined using the residual $\mathscr{R}(t)$,

$$E_g(t) := \mathscr{R}(t) = |\tilde{y}'_{m,p}(t) - f(t, \tilde{y}_{m,p}(t)|,$$

and its L_2-norm is defined by

$$\|E_g\|_2^2 = \int_{t_0}^{t_f} \mathscr{R}^2(t) dt \approx 2^{-m} \sum_i \mathscr{R}_i^2. \tag{5.8}$$

The derivative of the approximate solution $\tilde{y}'_{m,p}(t)$ is given by

$$\tilde{y}'_{m,p}(t) = \sum_{k=k_0-N+2}^{k_1-1} 2^m c_k \phi'(2^m t - k).$$

The values of $\phi'(t)$ at the integers $1, 2, \ldots, N-2$, are obtained in a similar way as described in section 2.4 for $\phi(t)$, with the coefficients h_k in (2.37) becomes $= 2h_k$ and the normalization condition (2.41) becomes $\sum_{l=1}^{N-2} \mu_{-l}^1 \phi'(l) = 1$.

The L^2-norm of $E_g(t)$, $\|E_g(t)\|_2^2$, is used as a measure of the global error over the interval $[t_0, t_f]$. We will use it to approximate the order of the methods. If the global error $E_g(t)$ of the method is $O(h^r)$, i.e., $E_g(t) = Kh^r$, where K is a constant that depends on t, we can use (5.7) or (5.8) to numerically approximate r as follows. Suppose $E_g(t) = Kh^r$. Then from (5.7), we have

$$\|E_g\|_2^2 = 2^{-m} \sum_i K_i^2 h^{2r} = \tilde{K} h^{2r+1}, \qquad \tilde{K} = \sum_i K_i^2, \qquad K_i = K(t_i).$$

Taking the logarithm, ln, we get

$$\ln(\|E_g\|_2^2) = \ln(\tilde{K}) + (2r+1)\ln(h).$$

Thus, the graph of $\ln(\|E_g\|_2^2)$ versus $\ln(h)$ is a straight line with slope $\alpha = (2r+1)$, from which we can solve for $r = \frac{\alpha-1}{2}$.

5.2 Linear Examples

In this section, we present the results obtained on different test IVPs.

Example 5.2.1. As a first example, we consider the simple linear problem

$$y' = -y, \qquad 0 \le t \le 2, \qquad y(0) = 1, \tag{5.9}$$

whose exact solution is $y_{exact}(t) = e^{-t}$.

We solve the problem (5.9) using the MWGMp-PY and the MWGMp-PF. The results are displayed in Figures 5.1- 5.4, where we plot the global error $E_g(t)$ versus t. It can be seen that $E_g(t)$ decreases as p and/or m increases, as expected. To numerically investigate the order of the methods, plots of $\|E_g(t)\|_2^2$ versus $h = 2^{-m}$ in $\log - \log$ scale for $p = 2,3,4,5$, are displayed in Figures 5.5-5.6. It could be seen that for lower p values ($p = 2,3,4$), the slope is approximately $2p-2$, from which the order r can be deduced to be $r \approx p - 3/2$, and hence we could numerically conclude that the global error is $O(h^{p-1})$. Numerical values for $\|E_g(t)\|_2^2$ for different values of p and m are displayed in Table 5.1.

Example 5.2.2. As a second example, we consider the linear problem

$$y' = \frac{2-ty}{t^2+1}, \qquad 0 \le t \le 1, \qquad y(0) = 1, \tag{5.10}$$

whose exact solution is $y_{exact}(t) = \frac{2t+1}{t^2+1}$.

For this example, we present the results for the MWGMpPY the results for MWGMp-PF are similar. The global error $E_g(t)$ for MWGMp-PY for different p and m values is displayed in Figures 5.7-5.8, where the $E_g(t)$ behavior is similar. Similarly, plots of $\|E_g(t)\|_2^2$ versus $h = 2^{-m}$ in $\log - \log$ scale for $p = 2,3,4,5$, are displayed in Figure 5.11. Again, we see that the slopes are approximately $2p-2$, from which the order r can be deduced to be $r \approx p - 3/2$, and hence we could numerically conclude that the global error is $O(h^{p-1})$.

5.3 Nonlinear Examples

In this section, we consider a nonlinear example to validate the accuracy of the MWGMp.

Example 5.3.1. Consider the nonlinear logistic model equation

$$y'(t) = ry(1-y/L), \qquad y(t_0) = y_0, \tag{5.11}$$

whose exact solution is $y_{exact}(t) = \frac{y_0 L}{y_0 + (L-y_0)e^{-rt}}$.

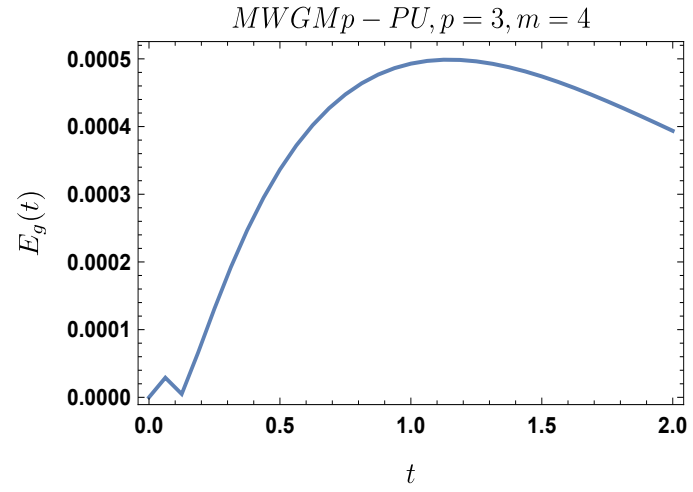
The parameters r and L are called the growth parameter and the carrying capacity, respectively. For numerical simulation, we consider the values $r = 10$ and $L = 4$, $y_0 = 1$, and $t_f = 1$. The exact solution is

$$y_{exact}(t) = \frac{4}{1+3e^{-10t}} \tag{5.12}$$

Plots of $E_g(t)$ with different p and m values for the MWGMp-PY are displayed in Figures 5.9-5.10. Similar numerical results are obtained when the MWGMp-PF is used.

Table 5.1: MWGMp-PY $\|E_g(t)\|_2^2$ for Example 5.2.1

$m\backslash p$	2	3	4	5
2	3.17947×10^{-3}	9.4268×10^{-5}	3.48638×10^{-6}	1.04859×10^{-7}
3	6.72203×10^{-4}	5.79421×10^{-6}	5.96729×10^{-8}	5.74887×10^{-10}
4	149645×10^{-4}	3.3401×10^{-7}	8.4798×10^{-10}	2.15185×10^{-12}
5	3.4914×10^{-5}	1.96725×10^{-8}	1.21584×10^{-11}	7.7829×10^{-15}
6	8.40462×10^{-6}	1.1877×10^{-9}	1.79941×10^{-13}	2.87817×10^{-17}
7	2.05997×10^{-6}	7.28651×10^{-11}	2.72781×10^{-15}	1.08916×10^{-19}
8	5.09802×10^{-7}	4.51051×10^{-12}	4.19475×10^{-17}	4.21585×10^{-22}
9	1.26799×10^{-7}	2.80533×10^{-13}	6.50064×10^{-19}	2.09616×10^{-24}
10	3.16182×10^{-8}	1.74902×10^{-14}	1.01078×10^{-20}	1.86961×10^{-25}



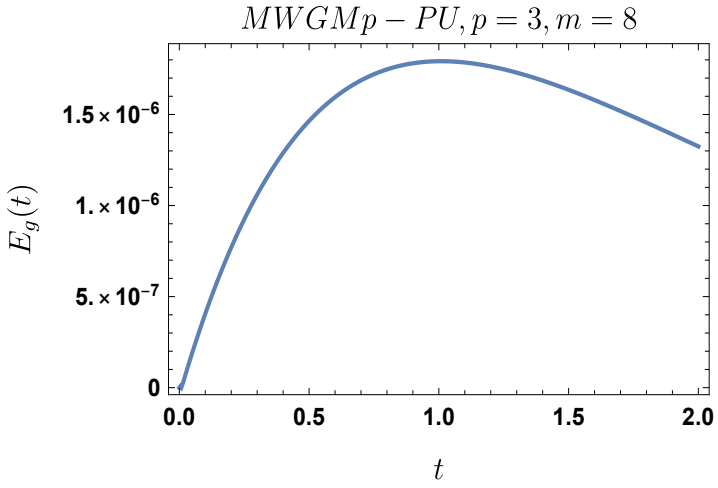


Figure 5.1: MWGMp-PY, $E_g(t)$ for Example 5.2.1, $p = 3$

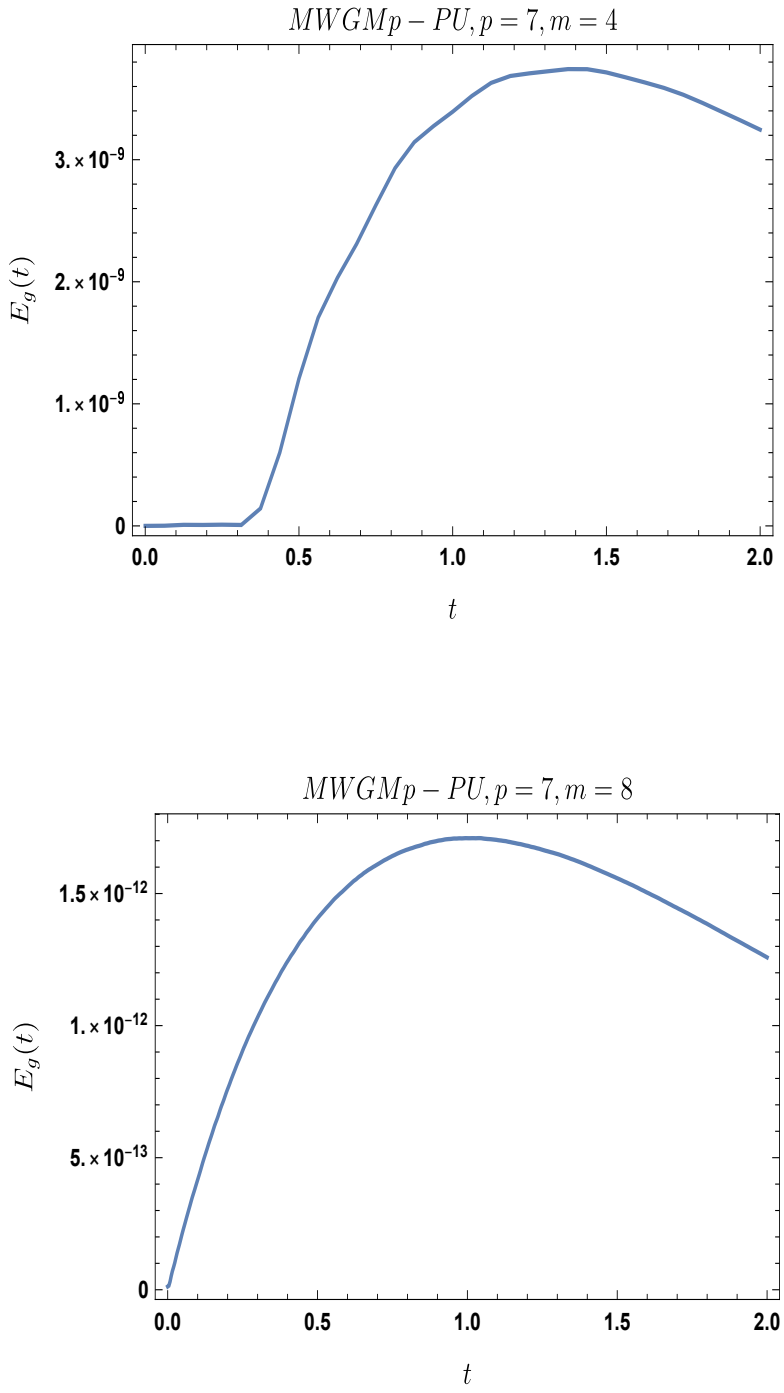


Figure 5.2: MWGMp-PY, $E_g(t)$ for Example 5.2.1, $p = 7$

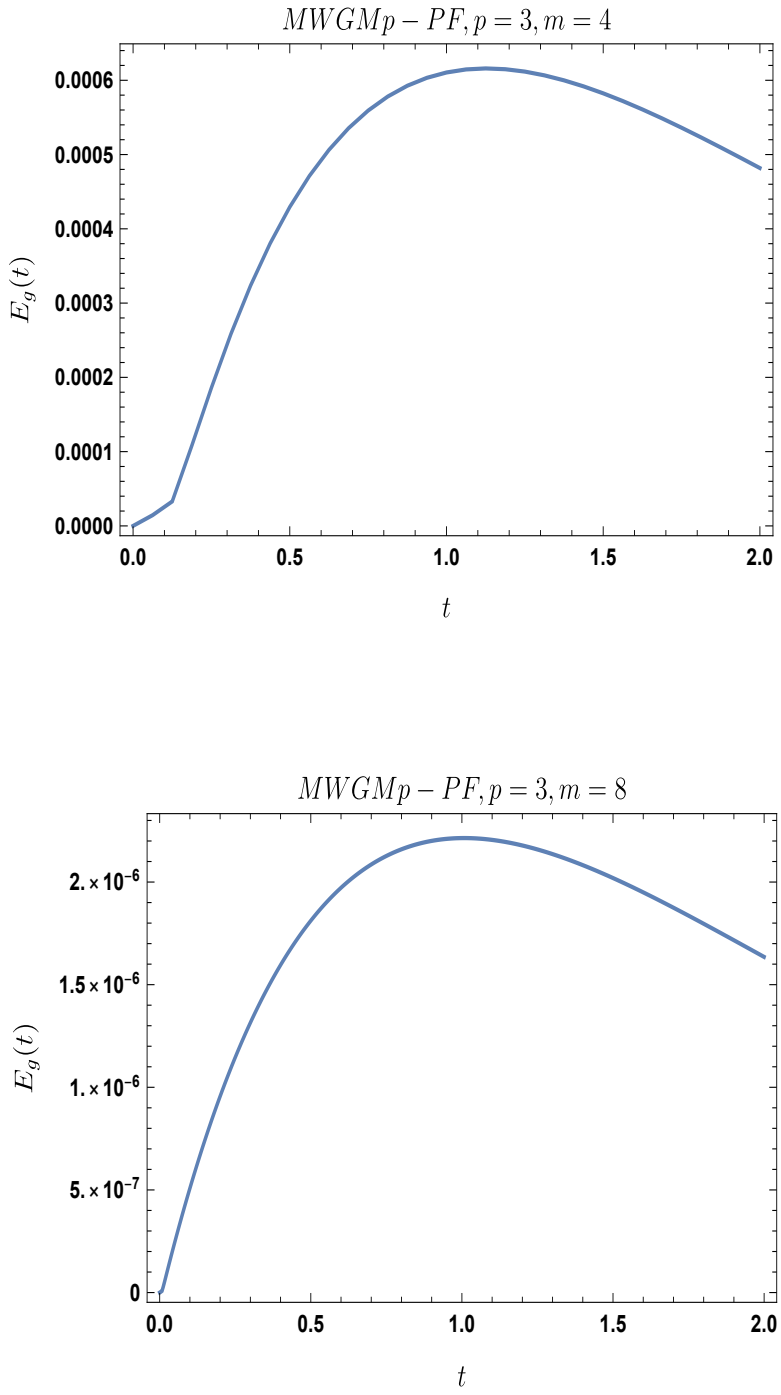


Figure 5.3: MWGMp-PF, $E_g(t)$ for Example 5.2.1, $p = 3$

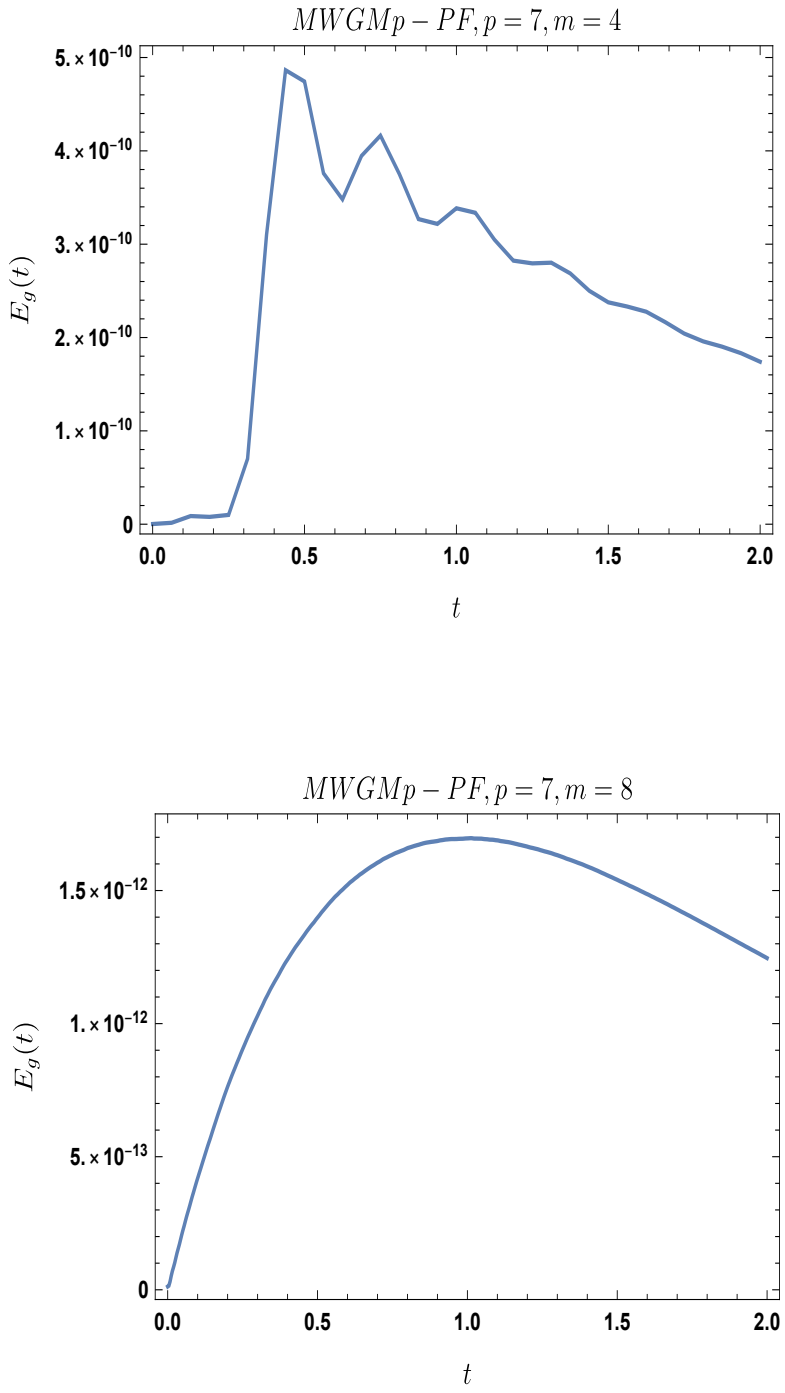


Figure 5.4: MWGMp-PF, $E_g(t)$ for Example 5.2.1, $p = 7$

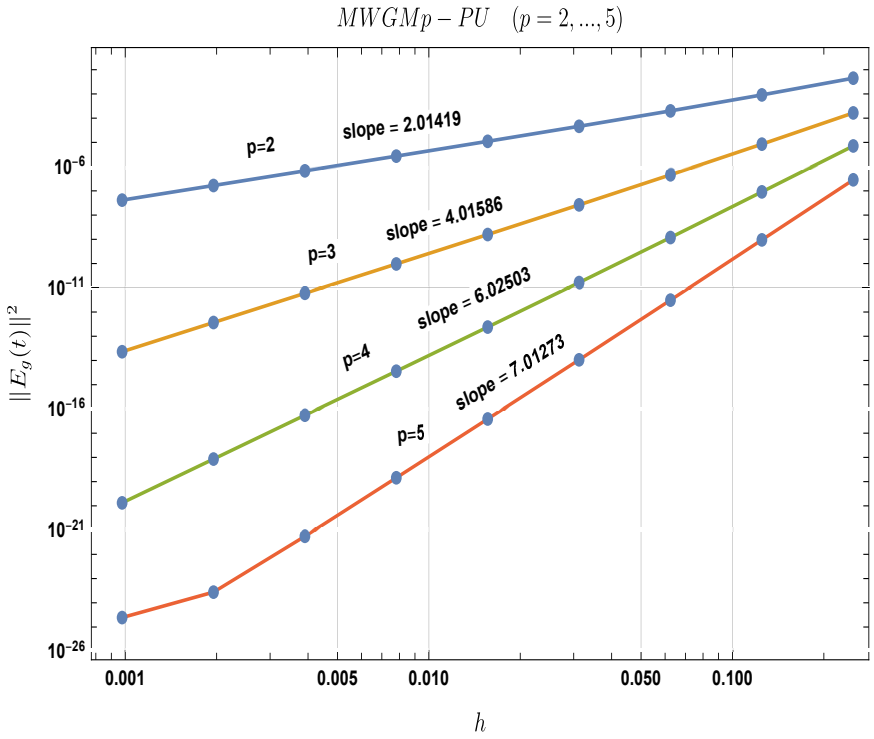


Figure 5.5: MWGMp-PY log – log plot of $\|E_g(t)\|_2^2$ vs h, Example 5.2.1

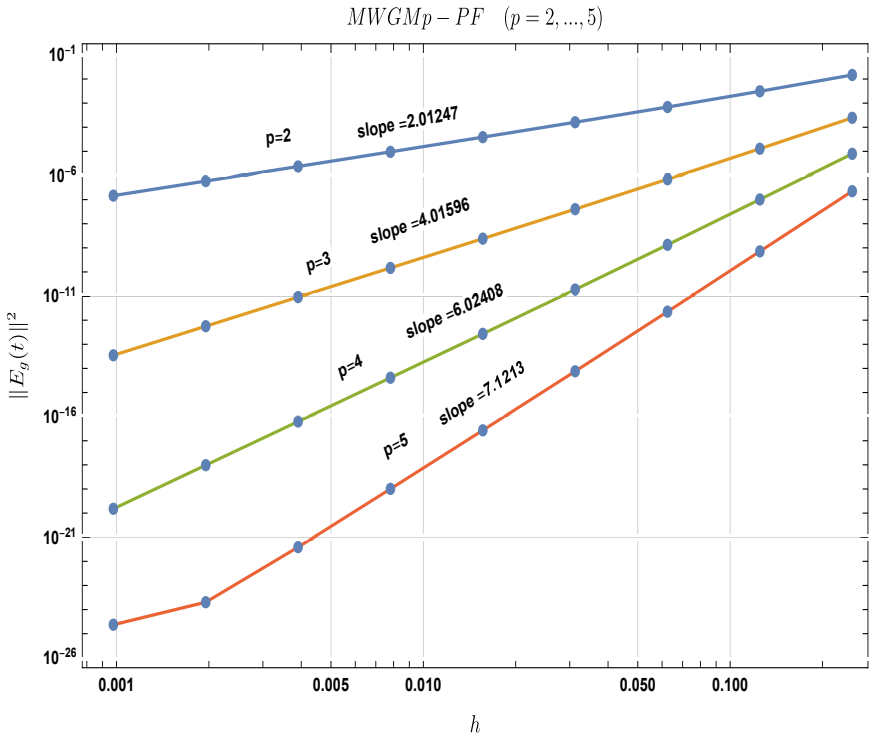


Figure 5.6: MWGMp-PF log – log plot of $\|E_g(t)\|_2^2$ vs h, Example 5.2.1

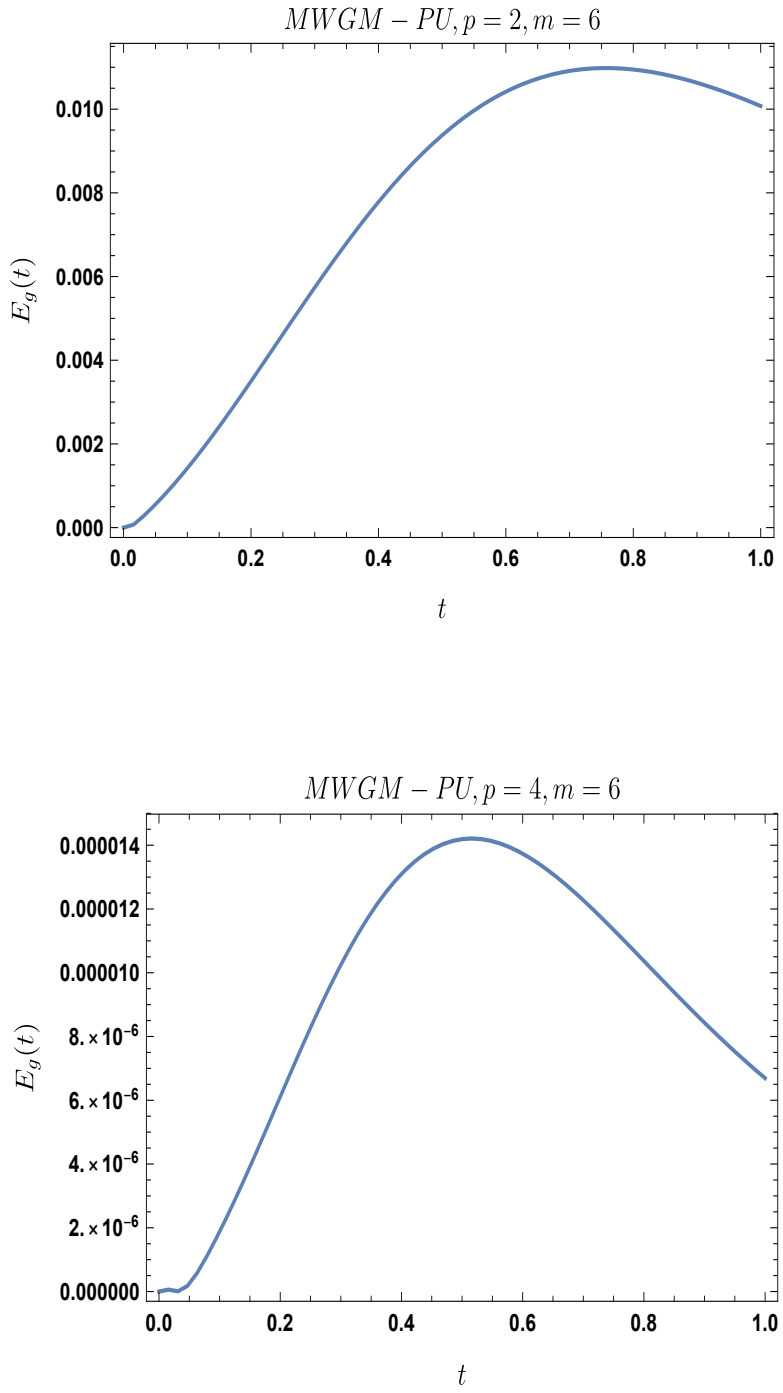


Figure 5.7: MWGMp-PY, $E_g(t)$ for Example 5.2.2, $m = 6$

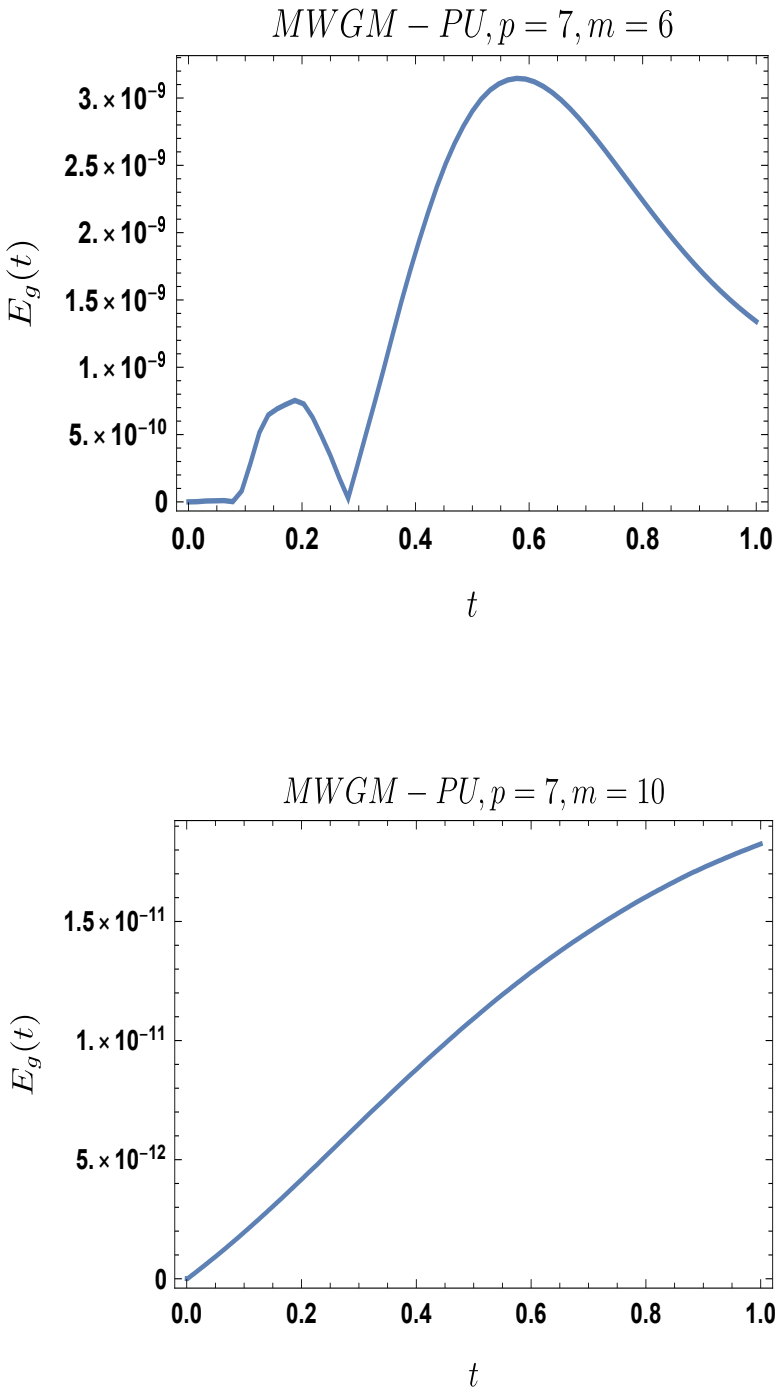


Figure 5.8: MWGMp-PY, $E_g(t)$ for Example 5.2.2, $p = 7$

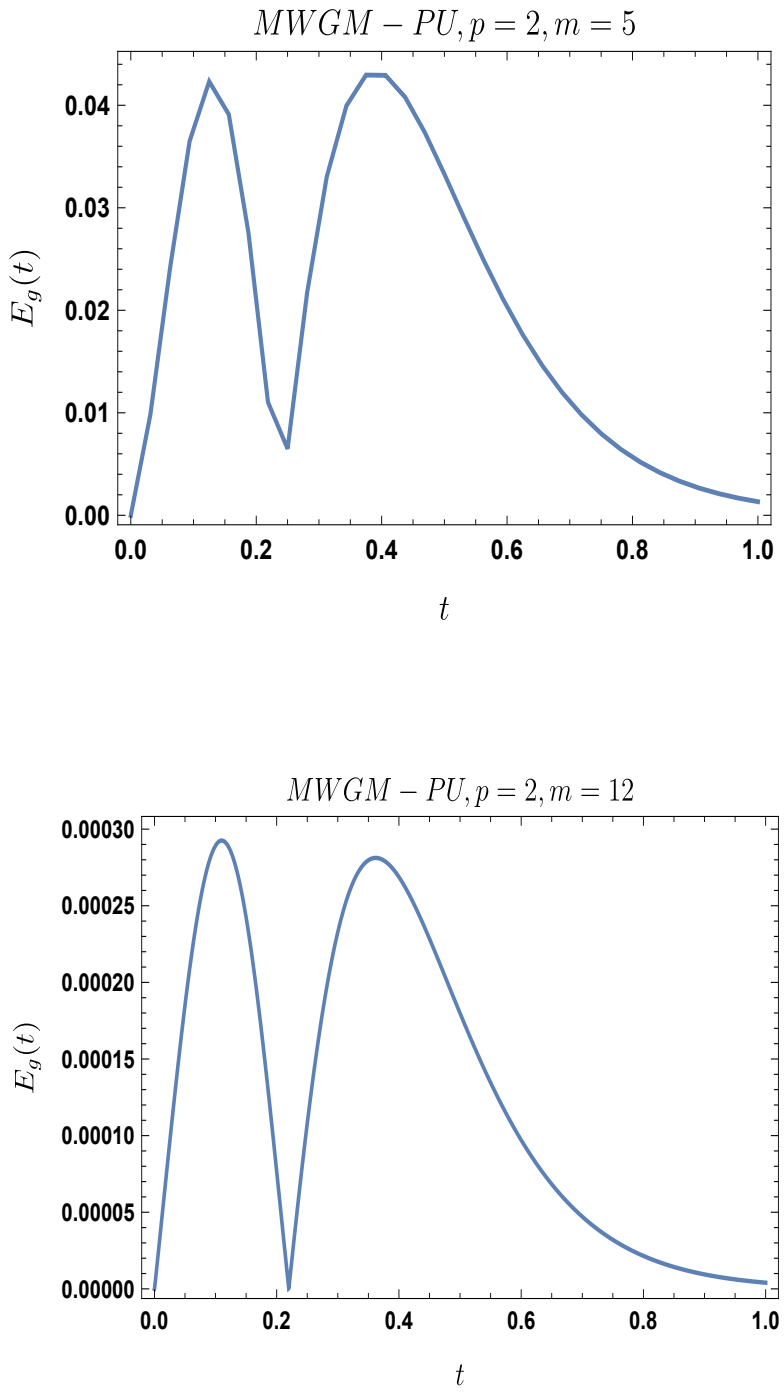


Figure 5.9: MWGMp-PY, $E_g(t)$ for Example 5.3.1, $p = 2$

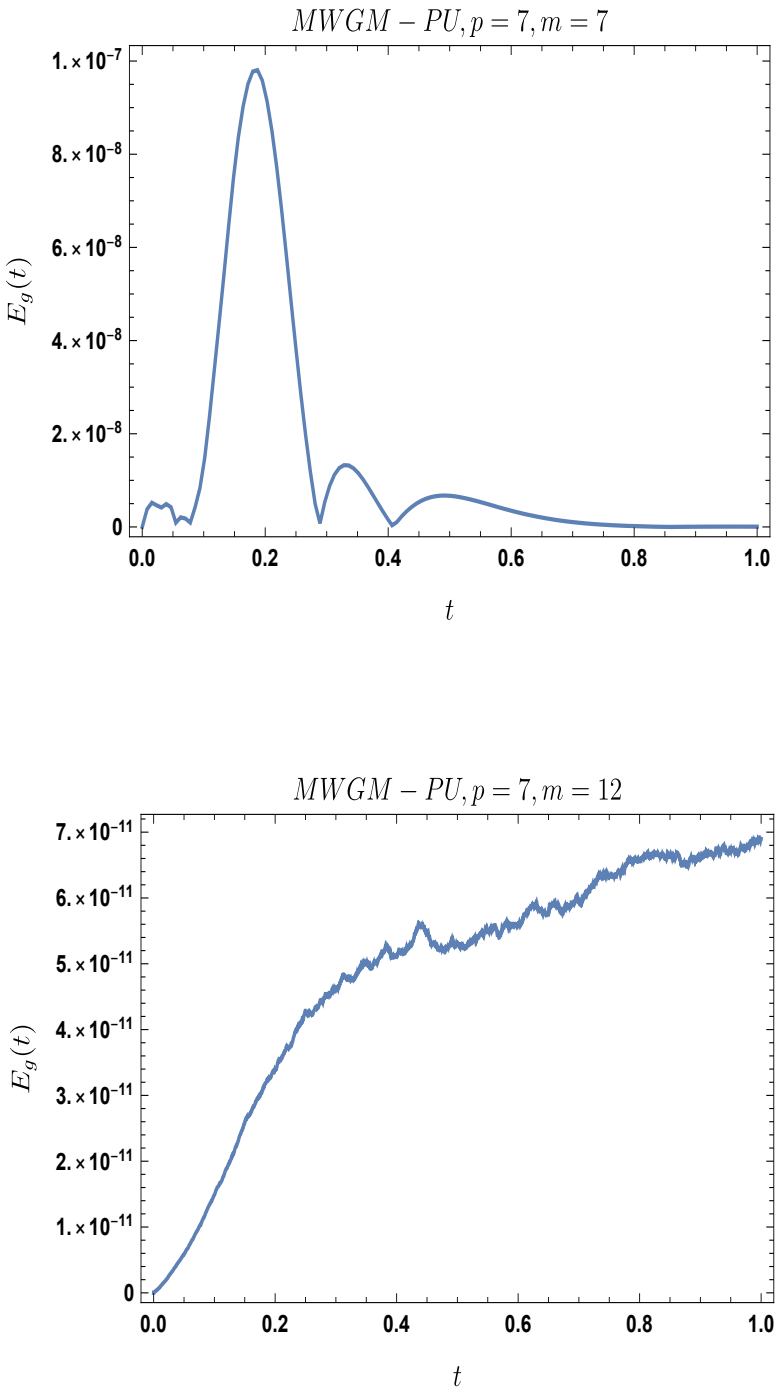


Figure 5.10: MWGMp-PY, $E_g(t)$ for Example 5.3.1, $p = 7$

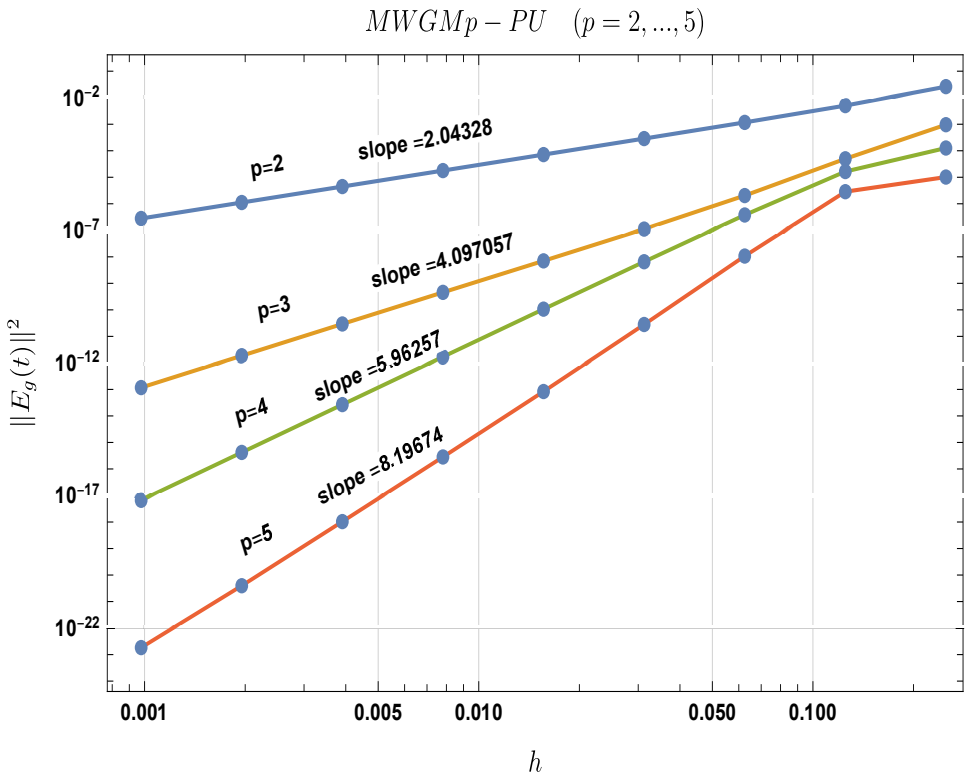


Figure 5.11: MWGMp-PY $\log-\log$ plot of $\|E_g(t)\|_2^2$ vs h, Example 5.2.2

Chapter 6: Conclusion

In this thesis, we employed Daubechies' wavelets to derive multistep methods for the numerical solutions of first-order initial value problems. Three schemes were derived: the MWGMp-PY, the MWGMp-PF, and the MWGMp-MP. It was observed that the MWGMp-PF and the MWGMp-MP are the same. The derivation was entirely based on the good approximation property of Daubechies' wavelets in that all polynomials of degree less than or equal to $p-1$ are exactly generated by integral translates of the scaling functions, where p is the order of the wavelet. This nice property makes the truncation error of the methods $O(h^p)$.

The stability analysis of the multistep methods were investigated and found that for $p=2$ and $p=3$ the methods are A-stable. The absolute stability regions of the derived multistep methods were found to be larger than those of Adam-Multon linear multistep methods for similar order. However, we found that Adam-Multon linear multistep methods have a better rate of convergence. The accuracy of the MWGMp-PY and MWGMp-PF was investigated by applying them to linear and nonlinear examples. The results show that they are competitive to Adam-Multon linear multistep methods.

A future research direction in this subject is to consider other wavelets such as coiflets whose scaling functions possess additional number of vanishing moments as well.

References

[1] Burden, R. L., & Faires, J. D. (2011). *Numerical Analysis*. Brooks/Cole, Cengage Learning.

[2] Haar, A. (1910). Zur Theorie der orthogonalen Funktionen systeme. *Math. Ann.*, 69, 331-371.

[3] Mallat, S. (1989). Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Trans. Amer. Math. Soc.*, 315, 69-88.

[4] Meyer, Y. (1992). *Wavelets and Operators*. Cambridge University Press, Cambridge.

[5] Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Comm. Pure and Appli. Math.*, 41, 909-996.

[6] Daubechies, I. (1992). *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia PA.

[7] Antoni, M., Barlaud, M., Mathieu, P., & Daubechies, I. (1992). Image coding using wavelet transform. *IEEE Trans. Image Proc.*, 205-220.

[8] Devore, R .A., Jawerth, B., & Lucier, B. J. (1992). Image compression through wavelet transform coding. *IEEE Trans. Inf. Th.*, 38, 719-746.

[9] Chen, C.F., & Hsiao C.H., (1999). Wavelet approach to optimizing dynamic systems. *IEEE Proc. Control Theory Appl.*, 146, 213-219.

[10] Lepik, U. (2008). Haar wavelet method for solving higher order differential equations. *International Journal of Mathematics and Computation*, 1, 84-94.

[11] Lepik, U. (2005). Numerical solution of differential equations using Haar wavelets. *Mathematics and Computers in Simulation,* 68, 127-143.

[12] Xu, J. C., & Shann, W. C. (1992). Galerkin-wavelet methods for two-point boundary value problems. *Numer.Math*, 36, 123-144.

[13] Vrscay, E. R. (1998). A Generalized Class of Fractal-Wavelet Transforms for Image Representation and Compression. *Canadian Journal of Electrical and Computer Engineering*, 23, 69-83.

[14] Coutsia, E. A., Hagstrom, T., & Torres, D. (1996). An efficient spectral method for ordinary differential equations with rational function coefficients. *Mathematics of Computation*, 65, 611-635.

[15] Fornberg, B. (1996). *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, Cambridge, UK.

[16] Canuto, C., Hussaini, M.Y., Quarteroni A., & Zang,T.A. (2006). *Spectral Methods, Fundamentals in Single Domains*. Springer-Verlag, Berlin Heidelberg.

[17] Gottlieb, D., & Orzag, S. (1977). *Numerical Analysis of Spectral Methods: Theory and Applications*. SIAM, Philadelphia, PA.

[18] Hesthaven, J., Gottlieb, S., & Gottlieb, D. (2007). *Spectral Methods for Time-dependent Problems*. Cambridge UP, Cambridge, UK.

[19] Trefethen, L. N. (2000). *Spectral Methods in MATLAB*. SIAM, Philadelphia, PA.

[20] Hariharan, G., Kannan, K., & Sharma, K. R (2009). Haar wavelet method for solving Fisher's equation. *Appl. Math. and Comput.*, 211, 284-292.

[21] Hariharan, G., & Kannan, K. (2010). Haar wavelet method for solving nonlinear parabolic equations. *J. Math. Chem.*, 48, 1044-1061.

[22] Sahu P. K., & Saha, R. S. (2017). Chebyshev wavelet method for numerical solutions of integro-differential form of Lane-Emden type differential equations. *International Journal of Wavelets, Multiresolution and Information Processing*, 15 .

[23] Manchanda, P., & Rani, M. (2017). Second Kind Chebyshev Wavelet Method for Solving System of Linear Differential Equations. *International Journal of Pure and Applied Mathematics*, 114, 91-104.

[24] Babolian, E., & Fattahzadeh, F. (2017). Numerical solution of differential equations by using Chebyshev wavelet operational matrix of integration. *Applied Mathematics and Computation*, 188, 417-426.

[25] Heydari, M. H., Hooshmandasl, M. R., & Maalek, F. M. (2014). A new approach of the Chebyshev wavelets method for partial differential equations with boundary conditions of the telegraph type. *Applied Mathematical Modelling*, 38, 1597-1606.

[26] Heil, C. E., & Walnut, D. F. (1989). Continuous and discrete wavelet transforms. *SIAM*, 31, 628-666.

[27] Chui, C. K. (1992). *An Introduction to Wavelets*. Academic Press, San Diego.

[28] Grossmann, A., Kronland-Martinet, R., & Morlet, J. (1989). Reading and understanding continuous wavelet transforms. *WAV89*, 2-20.

[29] Beylkin, G. (1992). On the representation of operators in bases of compactly supported wavelets.*SIAM J. Numer. Anal.*, 29, 1716-1740.

[30] Beylkin, G., Coifman, R., & Rokhlin, V. (1991). Fast wavelet transforms and numerical algorithms. *Comm. Pure Appl. Math.*, 44, 141-183.

[31] Beylkin, G., & Keiser, J. M. (1997). On the Adaptive Numerical Solution of Nonlinear Partial Differential Equations in Wavelet Bases. *Journal of Computational Physics*, 132, 233-259.

[32] Strang, G. & Nguyen, T. (1996). *Wavelets and Filter Banks*. Wellsley-Cambridge Press, Wellesley.

[33] Strang,G. (1989). Wavelets and dilation equations: A brief introduction. *SIAM Rev.*, 31, 614-627.

[34] Gray, R. M., & Goodman, J. W. (1995). *Fourier Transforms*. Kluwer.

[35] Bracewell, R. N. (1986). *The Fourier Transform and its Applications*. McGraw Hill.

[36] James, J. F. (2002). *A Student's Guide to Fourier Transforms*. Cambridge.

[37] Kammler, D. W. (2000). *A First Course in Fourier Analysis*. Prentice hall.

[38] Kemao, Q. (2004). Windowed Fourier transform for finger pattern analysis. *Applied Optics*, 43, 2695-2702.

[39] Cohen, L. (1995). *Time-frequency Analysis*. Prentice Hall, Englewood Cliffs (NJ).

[40] Hajji, M. A. (2016). On the Exact Values of Daubechies' Wavelets. *Journal of Physical Mathematics*, 7, 1-7.

[41] Baker,C. T. H., & Ford, N. J. (1991). Some applications of the boundary locus method and the method of D-partitions. *IMA J. Numer. Anal.*, 11, 143-158.

[42] Sanz-Serna, J. M. (1980). Some aspects of the boundary locus method. *BIT*, 20, 97-101.